



SODIMM-PLC7001

User Manual

Revision 1.01

Liability Exclusion

We have tested the contents of this document regarding agreement with the hardware and software described. Nevertheless, there may be deviations and we do not guarantee complete agreement. The data in the document is tested periodically, however. Required corrections are included in subsequent versions. We gratefully accept suggestions for improvements.

Copyright

Copyright © profichip GmbH 2009. All Rights Reserved. Unless permission has been expressly granted, passing on this document or copying it, or using and sharing its content are not allowed. Offenders will be held liable. All rights reserved, in the event a patent is granted or a utility model or design is registered.

This document is subject to technical changes.

1.1 Document Conventions

This document describes the SODIMM-PLC 7001 module.

CPU – Name:

- PLC 7001

CPU – Configuration:

- CPU 313-6CF03

Manufacturer: Profichip

In this User Description signal names appear *in italic*.

Low active signals are denoted by “#” after the signal name (i.e.: *RD#*).

A '0' indicates a logic-zero or low-level signal, while a '1' represents a logic-one or high-level signal.

1.2 Handling the Boards

Handle the SODIMM PLC 7001 module with care in respect to ESD requirements. Take precautions to use the boards only at an ESD-safe workplace.

1.3 User considerations

Objective and contents

This manual describes the the SODIMM-PLC 7001 module from Profichip. It contains a description of the construction, project implementation and usage.

Target audience

The manual is targeted at users who have a background in automation technology.

Structure of the manual

The manual consists of chapters. Every chapter provides a self-contained description of a specific topic.

Guide to the document

The following guides are available in the manual:

- an overall table of contents at the beginning of the manual
- an overview of the topics for every chapter
- an index at the end of the manual

Availability

The manual is available in electronic form as PDF-file (Adobe Acrobat Reader) only.

Icons Headings

Important passages in the text are highlighted by following icons and headings:



Danger!

Immediately or likely dangerous.
Personal injury is possible.



Attention!

Damages to property is likely if these warnings are not heeded.



Note!

Supplementary information and useful tips.

1.4 Safety information

Applications conforming with specifications

The SPEED7 CPU is constructed and produced for:

- communication and process control
- general control and automation applications
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle



Danger!

This device is not certified for applications in

- in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the



- project design department
- installation department
- commissioning
- operation

The following conditions must be met before using or commissioning the components described in this manual:

- Modification to the process control system should only be carried out when the system has been disconnected from power!
- Installation and modifications only by properly trained personnel
- The national rules and regulations of the respective country must be satisfied (installation, safety, EMC ...)

Disposal

National rules and regulations apply to the disposal of the unit!

SODIMM-PLC7001	1
1.1 Document Conventions	3
1.2 Handling the Boards	3
1.3 User considerations	4
1.4 Safety information	5
2 Hardware Description	9
2.1 Properties.....	10
2.2 Components	11
2.3 Digital In-/Output.....	13
2.4 Digital I/O – Parameterization	15
3 General Information	17
3.1 Safety Information for Users.....	17
3.2 Operating Structure of a CPU	18
3.3 CPU Applications.....	18
3.4 Operands of the CPU	19
3.5 CPU 313-6CF03	20
4 Basic Operations.....	23
4.1 Start-up Behavior.....	23
4.2 Initialization Ethernet PG/OP channel.....	24
4.2.1 Initialization via PLC functions	24
4.2.2 Initialization via minimal project.....	25
4.3 Access to the internal web page	26
4.4 Project engineering as CPU 313C-2DP.....	27
4.5 CPU parameterization	30
4.6 Operating SPEED7 200V modules	34
4.7 Project transfer	37
4.8 Operating modes	40
4.9 Overall reset	42
4.10 Firmware update.....	44
4.11 Factory reset.....	46
4.12 Memory extension with MCC	47
4.13 Extended know-how protection	48
4.14 MMC-Cmd - Auto commands.....	51
4.15 CPU specific diagnostic entries.....	52
4.16 Using test functions for control and monitoring of variables..	56
5 I/O periphery	59
5.1 Overview	59
5.2 In-/Output range.....	60
5.3 Digital part.....	60
5.4 Digital part – Parameterization	61
5.5 Counter - Brief Introduction	62
5.6 Counter - Controlling	66
5.7 Counter – Functions	71

5.7.1	Count Continuously	72
5.7.2	Count Once.....	72
5.7.3	Count Periodically.....	75
5.8	Counter - Additional functions	77
5.9	Counter - Diagnostic and interrupt	84
6	Dual Port Memory Interface.....	89
7	PtP Communication	93
7.1	Brief introduction.....	93
7.2	Protocols and procedures	94
7.3	Principals of the data transfer.....	98
7.4	RS485 interface	99
7.5	Communication.....	102
8	Profibus Communication.....	109
8.1	Overview	109
8.2	Profibus DP Slave.....	109
8.3	Profibus installation guidelines	111
8.4	Commissioning and Start-up behavior	115
9	Signal Description.....	116
9.1	MCU Interface (Dual-Port-RAM Interface)	118
9.2	VBUS Interface (Serial Extension Bus).....	120
9.3	Digital Inputs (DI)	120
9.4	Digital Outputs (DO)	121
9.5	Serial Interfaces (SI_0 - SI_3	121
9.6	Ethernet Interfaces (ETH1 & ETH2).....	122
9.7	FPGA Configuration Interface	122
9.8	Multi-Media-Card Interface (MMC).....	123
9.9	USB Interface	123
9.10	I ² C Interface	123
9.11	System Pins.....	124
9.12	General Purpose IOs (GPIO)	125
10	Electrical Characteristics	126
10.1.1	Absolute Maximum Ratings.....	126
10.1.2	Recommended Operating Conditions.....	126
10.1.3	DC Characteristics of Standard IOs.....	126
10.1.4	DC Characteristics of USB Transceiver.....	127
10.1.5	DC Characteristics of Ethernet PHY	127
11	External Circuit Description	129
11.1	SubD jack 9pin.....	129
11.2	RJ45 Ethernet jack	130
11.3	USB jack	130
11.4	MMC Card-slot.....	130
11.5	LEDs of the GPIO interface.....	130

Appendix	131
A – Index.....	131

2 Hardware Description

Overview

This chapter provides a general overview of the SODIMM module and information about the interfaces available at the SODIMM-PLC7001 with 313-6CF03 configuration. It describes functionality, project engineering and basic information of the digital IO.



Technical data

SODIMM-PLC 7001	
PLC Core CPU	PLC 7001
PLC Assembler Code	MC7
PLC Memory (Data/Code)	32kB / 32kB
Memory extension with MCC	max. 256kB / 256kB
System frequency	48MHz
Internal Cycle Time	20nsec.
System Interface	
User Data / Extension Port	DPR Interface (16Bit / 1024Byte)
On-Board I/Os	16DI / 16DO
Alarms (e. g. OB 40)	16
Counter (Up, Down, A/B, etc.)	3 x 32 Bit
Backplane-Bus, ser. I/O-Bus	x (max. 8 modules)
Communication Interfaces	
Ethernet 10/100	1x
MPI, 187.5 kBaud	1x
PROFIBUS-Slave, 12 MBit/s	1x
Standard Serial	1x
SD / MMC	1x
PLC System Counter and Timer	
Counter	512
Timer 10 msec.	512
High Resolution Timer 1μsec.	max
IEC Timer	max
RTC (+ext. Battery for Backup)	x
Operation Conditions	
Core Supply Voltage	5,0 V
I/O Voltage	3,3 V
Power Consumption	2,0 W
Temperature Range	-25°C - +60°C
Dimensions	
W x L x H (mm)	67,6 x 60,0 x 8,0

2.1 Properties

SODIMM-PLC7001

CPU 313-6CF03

- The SPEED7 PLC 7001 module is an embedded system
- Instruction set compatible to STEP[®]7 from Siemens
- Integrated DC5V power supply unit
- 64kByte total memory (32kByte code, 32kByte data)
- Memory expandable to max. 512kB (256kB code, 256kB data)
- MCC pins for external memory cards and memory extension
- Profibus DP slave
- Real-time clock
- Ethernet PG/OP interface integrated
- MPI interface
- RS485 interface configurable for PtP
- Digital I/Os: DI 16xDC3.3V, DO 16xDC3.3V
- 3 counter (30kHz)
- 512 timer
- 512 counter
- 8192 bit memory
- Serial I/O bus with up to 8 modules

Order Data

Type	Order number	Description
SODIMM PLC TYPE A	PA002100	DI 16xDC3.3V / DO 16xDC 3.3V, 3 counter with 64kByte (32kB/32kB) Data/Instruction RAM extendable up to 256kByte/256kByte with an MCC
SODIMM PLC TYPE B	PA002300	DI 16xDC3.3V / DO 16xDC 3.3V, 3 counter with 512kB (256kB/256kB) Data/Instruction RAM, memory <u>not</u> extendable

2.2 Components

CPU LEDs

The SODIMM-PLC 7001 has two status LEDs onboard to indicate power supply. However you can connect external status LEDs using the GPIO interface of the board. For detailed information please refer to the pin out assignment later in this document.

Storage Media Slot

As external storage medium for applications and firmware you may use a MMC (**M**ulti**M**edia **C**ard) or a MCC (**M**emory **C**onfiguration **C**ard) for memory extension. The MCC can additionally be used as an external storage medium.

Both storage media are pre-formatted with the PC format FAT and may be accessed via a card reader. An access to the storage media always happens after an overall reset and PowerON.

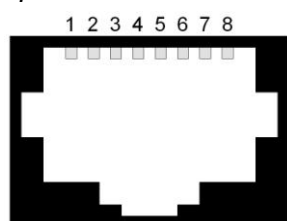
Ethernet PG/OP Channel

The ETH1 pins may be connected to a RJ45 jack to provide the interface to the Ethernet PG/OP channel. This interface allows you to program or remote control your CPU, to access the internal website or to connect visualization with up to 2 PG/OP connections. Here a transfer rate of 100MBit (full duplex) is supported.

For online access to the CPU via Ethernet PG/OP channel valid IP address parameters have to be assigned. More may be found at chapter "Deployment PLC 7001 (CPU 313-6CF03)" at "Initialization Ethernet PG/OP channel".

The jack has the following signal assignment:

8pin RJ45-slot:



Pin	Signal	Pin	Signal
1	Transmit +	5	-
2	Transmit -	6	Receive -
3	Receive +	7	-
4	-	8	-

Memory Management

The CPU has an integrated work memory. During program run the total memory is divided into 50% for program code and 50% for data.

There is the possibility to extend the total memory to its maximum by means of a MCC memory extension card.

Hardware Description

Service Interface SI0 – SI3

There are four Service interfaces integrated to the PLC. SI0 provides MPI with up to 187.5 MBit/s. SI1 is configured as PROFIBUS-Slave with 12 MBit/s maximum transfer rate. SI2 is for Standard serial communication like PtP connection. SI3 is not used in this configuration.

The interfaces have the following functionality:

	MPI	Profibus*	PtP
SI0	X		
SI1		X	
SI2			X
SI3	-	-	-

* Profibus DP Slave

In order to use this interfaces it is recommended to connect them to 9-pin SubD jacks to be compatible with standard systems.



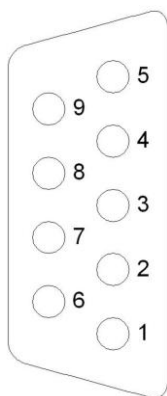
Note:

To support the SubD jack additional hardware is required.

Pin assignment

All interfaces have the same pin assignment:

9-pin SubD jack



Pin	Assignment
1	n.c.
2	M24V
3	RxD/TxD-P (line B)
4	RTS
5	M5V
6	P5V
7	P24V
8	RxD/TxD-N (line A)
9	n.c.

MPI functionality

The MPI interface handles the data exchange between CPU and PC. Via a bus communication you may transfer applications and data between the CPUs that are connected via MPI. Standard setting is MPI Address 2.

PtP functionality

With the PtP functionality the RS485 interface is allowed to connect via serial point-to-point connection to different source or target systems. The protocols ASCII, STX/ETX, 3964R, USS and Modbus master (ASCII, RTU) are supported.

The PtP communication is configured during run-time by means of the SFC 216 (SER_CFG). The communication happens by means of the SFC 217 (SER_SND) and SFC 218 (SER_RCV).

Profibus functionality

Using the *Profibus* functionality the integrated Profibus DP slave is connected to Profibus via RS485 interface. A slave will automatically adjust the parameters to the bus master.

2.3 Digital In-/Output

Overview

The SODIMM-PLC 7001 has the following digital in- and output ranges integrated:

- Digital Input: 16xDC 3,3V
- Digital Output: 16xDC 3,3V
- Technological functions: 3 Channels

Project Engineering

The project engineering takes place in the Siemens SIMATIC manager as CPU 313C-2DP from Siemens (6ES7 313-6CF03-0AB0 V2.0).

Here the CPU is parameterized by the "Properties" dialog of the CPU 313C-2DP.

For parameterization of the digital I/O periphery and the technological functions the corresponding sub module of the CPU 313C-2DP may be used.

By including the SPEED7.gsd the specific parameters of the PLC 7001 CPU may be set during hardware configuration. So for example the integrated RS485 interface may be parameterized.

I/O Periphery

The integrated I/Os of the PLC 7001 may be used for technological functions or as standard I/Os.

Technological functions and standard I/Os may be used simultaneously with appropriate hardware. Read access to inputs used by technological functions is possible. Write access to used outputs is not possible.

Hardware Description

Technological Functions

Up to 3 channels may be parameterized as technological function. The parameterization of the appropriate channel is made in the hardware configurator by the *count* sub module of the CPU 313C-2DP.

There are the following technological functions:

- Continuous count
- Single count
- Periodic count

The controlling of the corresponding counter mode happens by means of the SFB COUNT (SFB 47) of the user program.

Via the parameterization you may assign up to 16 on-chip alarm functions which results in low latency and fast response times upon critical system conditions and provides the capability to capture time critical events very accurately. Additionally the digital inputs can be parameterized as counter or may get interrupt properties assigned.

Access to the Digital Part

The PLC 7001 creates in its peripheral area an area for input respectively output data. Without a user specific hardware configuration the following default addresses are used.

Input Range

Sub module	Default Address	Access	Assignment
<i>DI16/DO16</i>	124	Byte	Digital Input I+0.0 ... I+0.7
	125	Byte	Digital Input I+1.0 ... I+1.7
<i>Count</i>	768	DInt	Channel 0: Count value / Frequency value
	772	DInt	Channel 1: Count value / Frequency value
	776	DInt	Channel 2: Count value / Frequency value
	780	DInt	reserved

Output range

Sub module	Default Address	Access	Assignment
<i>DI16/DO16</i>	124	Byte	Digital Output Q+0.0 ... Q+0.7
	125	Byte	Digital Output Q+1.0 ... Q+1.7
<i>Count</i>	768	DWord	reserved
	772	DWord	reserved
	776	DWord	reserved
	780	DWord	reserved

2.4 Digital I/O – Parameterization

Parameter Data

Parameters of the digital I/O may be set by means of the *DI16/DO16* sub module of the CPU 313C-2DP from Siemens during hardware configuration.

In the following paragraphs all parameters are specified, which may be used with the hardware configuration of the digital periphery.

General

This provides the short description of the digital periphery. At *Comment* information about the module such as purpose may be entered.

Addresses

At this register the start address of the in-/output periphery may be set.

Inputs

Here there are the following adjustment possibilities:

- Hardware interrupt
- Input delay

For the digital output channels there are no parameters.

Hardware Interrupt

A hardware interrupt may be optionally triggered on the rising or falling edge of an input. A diagnostic interrupt is only supported together with hardware interrupt lost.

Select with the arrow keys the input and activate the desired hardware interrupt.

Input Delay

The input delay may be configured per channel in groups of four. Please note that in the parameter window only the value 0.1ms may be set. At the other values 0.35ms is internally used for input delay.

3 General Information

3.1 Safety Information for Users



Handling of electrostatic sensitive modules

Profichip modules make use of highly integrated components in MOS-Technology. These components are extremely sensitive to over-voltages that can occur during electrostatic discharges.

The following symbol is attached to modules that can be destroyed by electrostatic discharges.



The Symbol is located on the module, the module rack or on packing material and it indicates the presence of electrostatic sensitive equipment.

It is possible that electrostatic sensitive equipment is destroyed by energies and voltages that are far less than the human threshold of perception. These voltages can occur where persons do not discharge themselves before handling electrostatic sensitive modules and they can damage components thereby, causing the module to become inoperable or unusable.

Modules that have been damaged by electrostatic discharges can fail after a temperature change, mechanical shock or changes in the electrical load.

Only the consequent implementation of protection devices and meticulous attention to the applicable rules and regulations for handling the respective equipment can prevent failures of electrostatic sensitive modules.

Shipping of modules

Modules must be shipped in the original packing material.

Measurements and alterations on electrostatic sensitive modules

When you are conducting measurements on electrostatic sensitive modules you should take the following precautions:

- Floating instruments must be discharged before use.
- Instruments must be grounded.

Modifying electrostatic sensitive modules you should only use soldering irons with grounded tips.



Attention!

Personnel and instruments should be grounded when working on electrostatic sensitive modules.

3.2 Operating Structure of a CPU

General

The CPU contains a processor with internal program memory.
A CPU supports the following modes of operation:

- cyclic operation
- timer processing
- alarm controlled operation
- priority based processing

Cyclic processing

Cyclic processing represents the major part of all the processes that are executed in the CPU. Identical sequences of operations are repeated in a never-ending cycle.

Timer processing

Where a process requires control signals at constant intervals you can initiate certain operations based upon a **timer**, e.g. not critical monitoring functions at one-second intervals.

Alarm controlled processing

If a process signal requires a quick response you would allocate this signal to an **alarm controlled** procedure. An alarm can activate a procedure in your program.

Priority based processing

The above processes are handled by the CPU in accordance with their **priority**. Since a timer or an alarm event requires a quick reaction, the CPU will interrupt the cyclic processing when these high-priority events occur to react to the event. Cyclic processing will resume, once the reaction has been processed. This means that cyclic processing has the lowest priority.

3.3 CPU Applications

Overview

The program that is present in every CPU is divided as follows:

- System routine
- User application

System Routine

The system routine organizes all those functions and procedures of the CPU that are not related to a specific control application.

User Application

This consists of all the functions that are required for the processing of a specific control application. The operating modules provide the interfaces to the system routines.

3.4 Operands of the CPU

Overview

The following series of operands is available for programming the CPU:

- Process image and periphery
- Bit memory
- Timers and counters
- Data blocks

Process Image and Periphery

The user application can quickly access the process image of the inputs and outputs PAA/PAE. You may manipulate the following types of data:

- individual Bits
- Bytes
- Words
- Double Words

You may also gain direct access to peripheral modules via the bus from user application. The following types of data are available:

- Bytes
- Words
- Blocks

Bit Memory

The bit memory is an area of memory that is accessible by means of certain operations. Bit memory is intended to store frequently used working data.

You may access the following types of data:

- individual Bits
- Bytes
- Words
- Double words

Timers and Counters

In your program you may load cells of the timer with a value between 10ms and 9990s. As soon as the user application executes a start-operation, the value of this timer is decremented by the interval that you have specified until it reaches zero.

You may load counter cells with an initial value (max. 999) and increment or decrement these when required.

Data Blocks

A data block contains constants or variables in the form of bytes, words or double words. You may always access the current data block by means of operands.

You may access the following types of data:

- individual Bits
- Bytes

- Words
- Double words

3.5 CPU 313-6CF03

Overview

The PLC 7001 is based upon the SPEED7 technology. This supports the CPU at programming and communication by means of co-processors that causes a power improvement for highest needs.

The CPU is programmed in STEP[®]7 from Siemens. For this you may use WinPLC7 from VIPA or the Siemens SIMATIC Manager.

The CPU with integrated Ethernet-PG/OP channel, a MPI-, Profibus-Slave and RS485-slot simplifies the integration of the CPU into an existing network or the connection of additional peripheral equipment.

Memory Management

The CPU has an integrated work memory. During program run the total memory is divided into 50% for program code and 50% for data.

There is the possibility to extend the total memory to its maximum by means of a MCC memory extension card.

Integrated Ethernet-PG/OP-Channel

The CPU has an Ethernet interface for PG/OP communication. After the assignment of IP address parameters by "Assign Ethernet Address" respectively by a "minimum project" the Ethernet PG/OP channel may directly be addressed by means of the "PLC" functions to program and remote control the CPU. A maximum of 2 PG/OP connections is available.

You may also access the CPU with visualization software via these connections.

Integrated Profibus DP Slave

The CPU has an integrated Profibus DP slave. The project engineering takes place in WinPLC7 from VIPA or in the hardware configurator from Siemens.

Environmental Conditions

- Operating temperature: 0 ... +60°C
- Storage temperature: -25 ... +70°C
- Relative humidity: 5 ... 95% without condensation
- Ventilation by means of a fan is not required

Compatibility

The CPU 313-6CF03 is able to operate up to 8 external modules which can be configured with the Siemens hardware configurator using the SPEED7.gsd file. Further information may be found in "Operating SPEED7 200V modules".



Note!

Please do always use the **CPU 313C-2DP (6ES7 313-6CF03-0AB0 V2.0)** from Siemens from the hardware catalog to project this SODIM-PLC 7001 version from Profichip.

For the project engineering, a thorough knowledge of the Siemens SIMATIC Manager and the hardware configurator from Siemens is required!

Power Supply

The SODIMM module requires a power supply of 5V DC.

4 Basic Operations



Note!

All the subsequent chapters refer to the SODIMM-PLC7001 configured as a **CPU 313-6CF03**! Other configurations are currently not supported!

4.1 Start-up Behavior

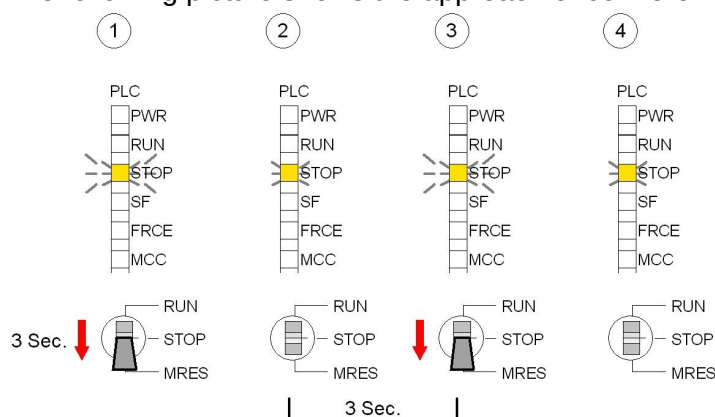
Turn on power supply

After the power supply has been switched on, the CPU changes to the operating mode indicated by the operating mode lever.

Now you may transfer your project to the CPU via MPI from your configuration tool or plug in a MMC with your project and execute an overall reset.

Overall reset

The following picture shows the approach once more:



Note!

The transfer of the application program from the MMC into the CPU takes always place after an overall reset!

Delivery status

When the CPU is delivered it has been reset. After a STOP→RUN transition the CPU switches to RUN without program.

Boot procedure with valid data in the CPU

The CPU switches to RUN with the program stored in the battery buffered RAM.

Boot procedure with empty battery

The accumulator/battery is automatically loaded via the integrated power supply and guarantees a buffer for max. 30 days. If this time is exceeded, the battery may be totally discharged. This means that the battery buffered RAM is deleted.

In this state, the CPU executes an overall reset. If a MMC is plugged, program code and data blocks are transferred from the MMC into the work memory of the CPU.

If no MMC is plugged, the CPU transfers permanent stored "protected" blocks into the work memory if available.

Information about storing protected blocks in the CPU can be found in this chapter at "Extended Know-how protection".

Depending on the position of the RUN/STOP lever, the CPU switches to RUN res. remains in STOP.

This event is stored in the diagnostic buffer as: "Start overall reset automatically (unbuffered POWER_ON)".

4.2 Initialization Ethernet PG/OP channel

Overview

The SPEED7 PLC 7001 has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU with up to 2 connections.

The PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc.

For online access to the CPU via Ethernet PG/OP channel valid IP address parameters have to be assigned to this by means of the Siemens SIMATIC manager. This is called "Initialization".

Possibilities for Initialization

There are the following possibilities for assignment of IP address parameters (initialization):

- PLC functions with *Assign Ethernet address*
- Hardware project engineering with CP (Minimal project)

Requirements

For the hardware configuration the following software is necessary:

- SIMATIC Manager from Siemens V. 5.1 or higher
- SIMATIC NET

4.2.1 Initialization via PLC functions

The initialization takes place after the following proceeding:

- Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel.
- Establish a network connection between Ethernet PG/OP channel of the CPU and PC.
- Start the Siemens SIMATIC manager at the PC.
- Set via **Options** > *Set PG/PC Interface* the Access Path to "TCP/IP -> Network card Protocol RFC 1006".
- Open with **PLC** > *Assign Ethernet Address* the dialog window for "initialization" of a station.

- Use the [Browse] button to determine the CPU components via MAC address. As long as the Ethernet PG/OP channel was not initialized yet, this owns the IP address 0.0.0.0 and the station name "Onboard PG/OP".
- Choose the determined module and click to [OK].
- Set the IP configuration by entering IP address, subnet mask and net transition. In addition an IP address may be received from a DHCP server. For this depending upon the selected option the MAC address, device name or the Client ID, which may be entered here, is to be conveyed to the DHCP server. The Client-ID is a character sequence from maximally 63 characters. Here the following indications may be used: Dash "-", 0-9, a-z, A-Z
- Confirm your settings by button [Assign Address]

After the assignment the Ethernet PG/OP channel may be reached by the Siemens SIMATIC manager by means of these IP address parameters and the *Access Path* "TCP/IP -> Network card Protocol RFC 1006".

4.2.2 Initialization via minimal project

- Establish a network connection between Ethernet PG/OP channel of the CPU and PC.
- Start the SIMATIC Manager from Siemens and create a new project.
- Add a new System 300 station via **Insert** > *Station* > *SIMATIC 300-Station*.
- Activate the station "SIMATIC 300" and open the hardware configurator by clicking on "Hardware".
- Engineer a rack (SIMATIC 300 \ Rack-300 \ Profile rail)
- Place the Siemens CPU 312C with the order no. 6ES7 312-5BE03-0AB0 V2.0 from the hardware catalog. This may be found at SIMATIC 300 \ CPU 300 \ CPU 312C.
- Include the CP 343-1EX11 at slot 4 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1).
- Type the wanted IP address and subnet mask into the dialog window of "Properties" of the CP 343-1 and connect the CP with "Ethernet".
- Save and compile your project.
- Transfer your project via MPI or MMC into your CPU. More information about transfer methods may be found in the chapter "Project transfer".

Direct after the assignment the Ethernet PG/OP channel may be reached by the Siemens SIMATIC manager by means of these IP address parameters and the *Access Path* "TCP/IP -> Network card Protocol RFC 1006".

4.3 Access to the internal web page

Access to the web page

The Ethernet PG/OP channel provides a web page that you may access via an Internet browser by its IP address. The web page contains information about firmware versions, current cycle times etc. The current content of the web page is stored on MMC by means of the MMC-Cmd WEBPAGE. More information may be found at "MMC-Cmd - Auto commands".

Requirements

A PG/OP channel connection should be established between PC with Internet browser and CPU 313 6CF03. This may be tested by *Ping* to the IP address of the Ethernet PG/OP channel.

Web page



The access takes place via the IP address of the Ethernet PG/OP channel. The web page only serves for information output. The monitored values are not alterable.

```
CPU WITH ETHERNET PG/OP
Slot 100
7001 SODIMM V3.4.2 Px000086.pkg,
SERIALNUMBER 02106
SUPPORTDATA: PRODUCT V3118, HARDWARE ...
Memoriesizes (Bytes): LoadMem : 65536,
WorkMemCode : 32768, WorkMemData : 32768
OnBoardEthernet : MacAddress :
0020d577083A, IP-Address : , SubnetMask : ,
Gateway :
Cpu state : RUN
FunctionRS485 X2: MPI
FunctionRS485 X3/COM2: PtP
Cycletime [microseconds] : min=17000
cur=17000 ave=17000 max=17000
```

MCC-Trial-Time: 70:23

```
Slot 202
7001 DI16/DO16 V3.4.2 , SUPPORTDATA :
PRODUCT V3422
SUPPORTDATA : PRODUCT V3422, Module Type
...
Address Input 124...125
Address Output 124...125
Slot 204
3 COUNTERS V3.4.2

SUPPORTDATA : PRODUCT V3422, Module Type
```

Order no., firmware vers., package,
serial no.
Information for support

Ethernet PG/OP: Addresses

CPU state
RS485 function of X2
RS485 function of X3
CPU cycle time:
min= minimal, cur= current
ave= average, max= maximal

Remaining time for deactivation of
the expansion memory if MCC is
removed.

Additional CPU components:
Slot 202 (Digital I/Os):
Name, firmware version, module
type
Information for support

Configured input base addresses
Configured output base addresses
Slot 204 (Counter)
Name, firmware version, module
type
Information for support

...

Address Input 768...783
Address Output 768...783

Configured input base addresses
Configured output base addresses

Standard Bus BaudRate Read Model , BaudRate
Write Model

Modules at standard bus

4.4 Project engineering as CPU 313C-2DP

Overview

The project engineering of the PLC 7001 takes place in the Siemens hardware configurator and is divided into the following parts:

- Project engineering PLC 7001 as CPU 313C-2DP from Siemens (6ES7 313-6CF03-0AB0 V2.0).
- Project engineering Ethernet PG/OP channel as CP 343-1 (343-1EX21-0XE0).
- Project engineering of Speed7 200V modules at the bus.

Requirements

The hardware configurator is a part of the Siemens SIMATIC Manager and is used for the project engineering. The modules that may be configured here are listed in the hardware catalog. If necessary you have to update the hardware catalog with **Options** > *Update Catalog*.

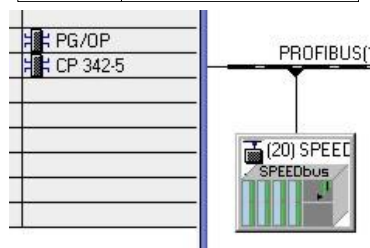
For the project engineering a thorough knowledge of the Siemens SIMATIC Manager and the hardware configurator from Siemens are required and assumed!

Basic Operations

Fast introduction

To be compatible with the Siemens hardware configurator the following steps should be executed:

Slot	Module
1	
2	CPU 313C-2 DP
X2	DP
2.2	DI16/DO16
2.4	Count
4	343-1EX21-0XE0
5	342-5DA02-0XE0



- Start the hardware configurator from Siemens.
- Configure CPU 313C-2DP (6ES7 313-6CF03-0AB0 V2.0) from Siemens.
- For the internal Ethernet PG/OP channel that is integrated to every SC CPU, you have to configure a Siemens CP 343 -1 (343-1EX21). Let at *options* the attitude "Save configuration data on the CPU" activated! As there are no other modules this is placed at slot 4.
- Add a CP 342-5 Module (6GK7 342-5DA02-0XE0 V5.0). Access the configuration of this module and set [Operating Mode] DP-Master.
- Place a SPEED7 module at the profibus which work as rack for Speed7 200V modules. (with addresses 20)
- Add your Speed7 200V modules to the SPEED7 module (max. 8).

Steps of the project engineering

The project engineering is separated into 3 parts:

- Project engineering of the CPU
- Project engineering of the PG/OP channel
- Project engineering of Speed7 200V modules

Project engineering CPU as CPU 313C-2DP

- Start the hardware configurator from Siemens with a new project and insert a profile rail from the hardware catalog.
- Place the following Siemens CPU at slot 2:
CPU 313C-2DP (6ES7 313-6CF03-0AB0 V2.0)

Project engineering of Ethernet PG/OP channel as 343-1EX21

For the internal Ethernet PG/OP channel you have to configure a Siemens CP 343-1 (343-1EX21) module. This may be found at the hardware catalog at SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 **343-1EX21 0XE0**.

In the configuration of CP343-1 in the tab [*options*] the attitude "Save configuration data on the CPU" must be activated!

Slot	Module
1	
2	CPU 313C-2 DP
X2	DP
2.2	DI16/DO16
2.4	Count
4	343-1EX21-0XE0
5	342-5DA02-0XE0

Set IP parameters

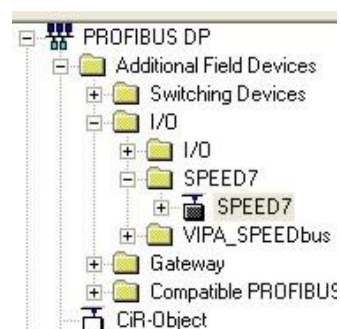
Open the property window via double-click on the CP 343-1EX21. Enter "General" and click at [Properties]. Type in the *IP address*, *subnet mask* and *gateway* for the CP and select the wanted *subnet*.

Project engineering of Speed7 200V modules

The PLC 7001 CPU is able to operate with Speed7 200V modules. To configure this with the STEP7 HW Config you need a CP 342-5 Module added to your rack which is located in **SIMANTIC 300 \ CP-300 \ PROFIBUS \ CP 342-5 \ 6GK7 342-5DA02-0XE0**. Access the configuration of this module and set [Operating Mode] DP-Master.

You can now attach a Speed7 module to this profibus which work as rack for Speed7 200V modules and is located in PROFIBUS DP \ Additional Field Devices \ I/O \ SPEED7.

The Speed7 module must have profibus-address set to 20.
The Speed7 module supports up to 8 Speed7 200V.



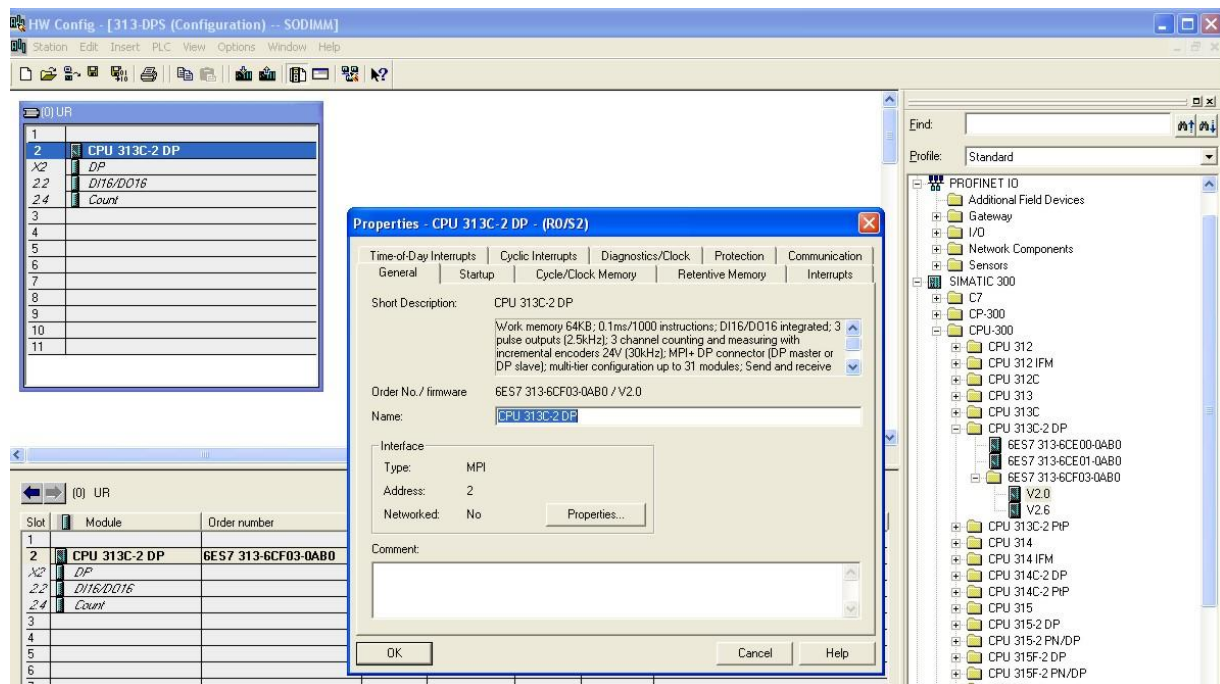
4.5 CPU parameterization

Overview

Since the **CPU** – Configuration CPU 313-6CF03 of PLC 7001 is to be configured as Siemens CPU 313C-2DP in the Siemens hardware configurator, the parameters of the SPEED7 PLC 7001 may be set with "Object properties" during hardware configuration.

Via a double click on the CPU 313C-2DP the parameter window may be accessed.

Using the registers you get access to all parameters of the CPU.



Note!

A description of the parameters of the sub module *DI16/DO16* and *Count* may be found at chapter "Deployment I/O periphery".

Supported parameters

The CPU does not evaluate all parameters that may be set at the hardware configuration. The following parameters are supported at this time:

General

Short description

Since the PLC 7001 (with CPU 313-6CF03 configuration) is configured as CPU 313C-2DP from Siemens, here the short description CPU 313C-2DP stands.

Order No. / Firmware

Order number and firmware are identical to the details in the "Hardware catalog" window.

Name

The *Name* field provides a short description of the module, which you can change to meet your requirements. If you change the description, the new description appears in the SIMATIC Manager.

Interface

Here the address of the MPI interface stands.

Properties

Click the "Properties" button to change the properties of the MPI interface.

Comment

In this field information about the module may be entered.

Startup

Startup when expected/actual configuration differs

If the checkbox for "Startup when expected/actual configuration differ" is *deselected* and at least one module is not located at its configured slot or if another type of module is inserted there instead, then the CPU switches to STOP mode.

If the checkbox for "Startup when expected/actual configuration differ" is *selected*, then the CPU starts even if there are modules are not located in their configured slots or if another type of module is inserted there instead, such as during an initial system start-up.

Monitoring Time for Ready message by modules [100ms]

This operation specifies the maximum time for the ready message of all configured modules after PowerON. If the modules do not send a ready message to the CPU by the time the monitoring time has expired, the actual configuration becomes unequal to the preset configuration.

Monitoring Time for Transfer of parameters to modules [100ms]

The maximum time to transfer the configuration to the parameterized modules. If not all of the modules have been assigned parameters by the time this monitoring time has expired, the actual configuration becomes unequal to the preset configuration.

Cycle/Clock memory

Scan Cycle Monitoring Time

Here the scan cycle monitoring time in milliseconds may be set. If the scan cycle time exceeds the scan cycle monitoring time, the CPU enters the STOP mode. Possible reasons for exceeding the time are:

- Communication processes
- a series of interrupt events
- an error in the CPU program

Basic Operations

Scan Cycle Load from Communication

Using this parameter you can control the duration of communication processes, which always extend the scan cycle time so it does not exceed a specified length.

If there are no additional asynchronous events, the scan cycle time of OB1 is increased by following factor:

$$\frac{100}{100 - \text{Cycle load from communication \%}}$$

If the cycle load from communication is set to 50%, the scan cycle time of OB 1 can be doubled. At the same time, the scan cycle time of OB 1 is still being influenced by asynchronous events (e.g. process interrupts) as well.

OB85-Call up at I/O Access Error

The preset reaction of the CPU may be changed to an I/O access error that occurs during the update of the process image by the system.

The CPU 313 6CF03 is preset such that OB 85 is not called if an I/O access error occurs and no entry is made in the diagnostic buffer either.

Clock Memory

Activate the check box if you want to use clock memory and enter the number of the memory byte.



Note!

The selected memory byte cannot be used for temporary data storage.

Retentive Memory

Number of Memory Bytes from MB0

Enter the number of retentive memory bytes from memory byte 0 onwards.

Number of S7 Timers from T0

Enter the number of retentive S7 timers from T0 onwards. Each S7 timer occupies 2 bytes.

Number of S7 Counters from C0

Enter the number of retentive S7 counter from C0 onwards.

Interrupts

Hardware Interrupts

Currently, the default priority may not be modified.

Time-of-Day interrupts

Priority

The priority may not be modified.

Active

Activate the check box of the time-of-day interrupt OBs if these are to be automatically started on complete restart.

Execution

Select how often the interrupts are to be triggered. Intervals ranging from every minute to yearly are available. The intervals apply to the settings made for *start date* and *time*.

Start date / Time

Enter date and time of the first execution of the time-of-day interrupt.

Process image partition

Is not supported.

Cyclic interrupts

Priority

The preset priority may not be modified.

Execution

Enter the time intervals in ms, in which the watchdog interrupt OBs should be processed. The start time for the clock is when the operating mode switch is moved from STOP to RUN.

Phase Offset

Not adjustable.

Process image partition

Is not supported.

Protection

Level of protection

Here 1 of 3 protection levels may be set to protect the CPU from unauthorized access.

Protection level 1 (default setting):

- No password adjustable, no restrictions

Protection level 2 with password:

- Authorized users: read and write access
- Unauthorized user: read access only

Protection level 3:

- Authorized users: read and write access
- Unauthorized user: no read and write access

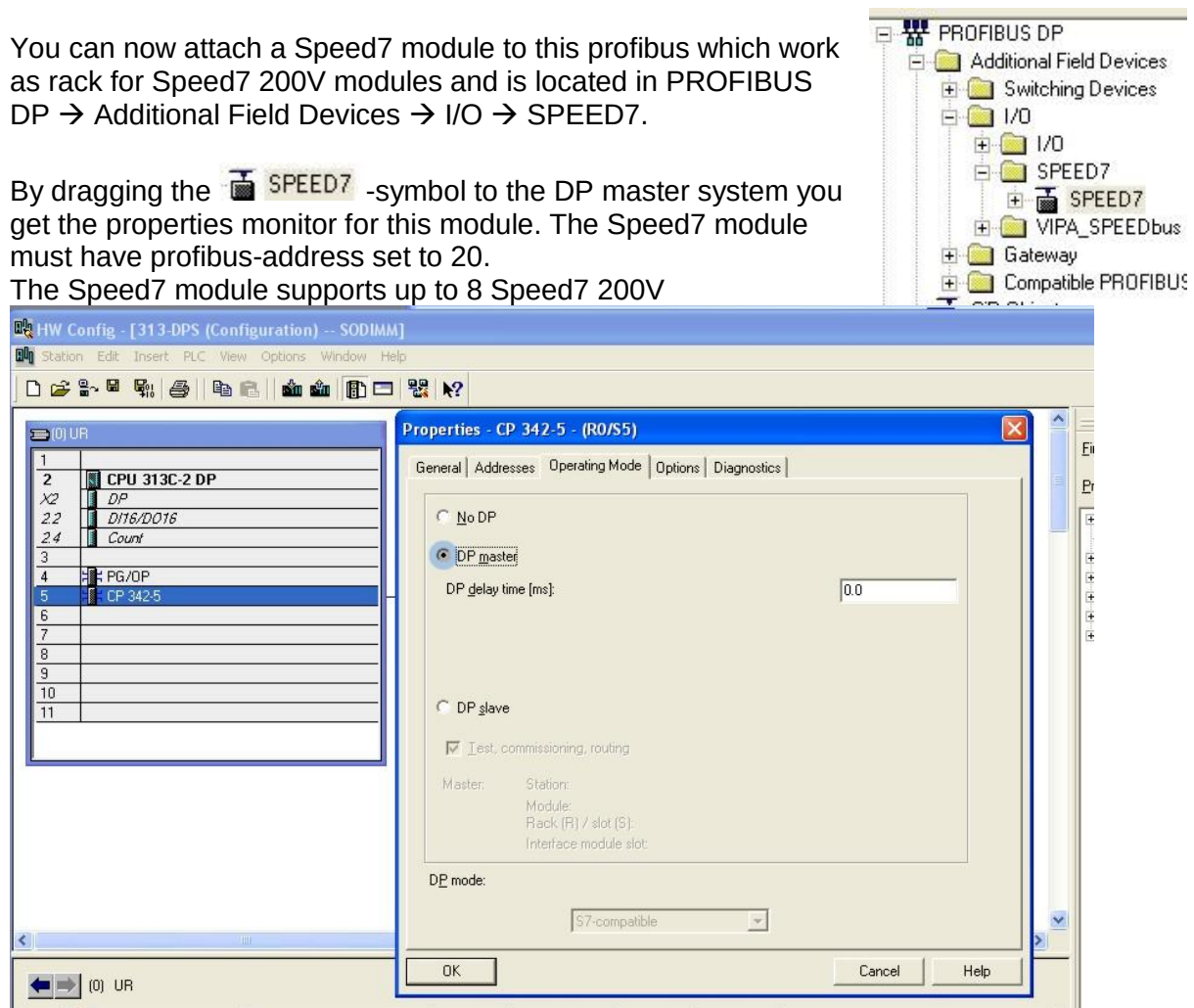
4.6 Operating SPEED7 200V modules

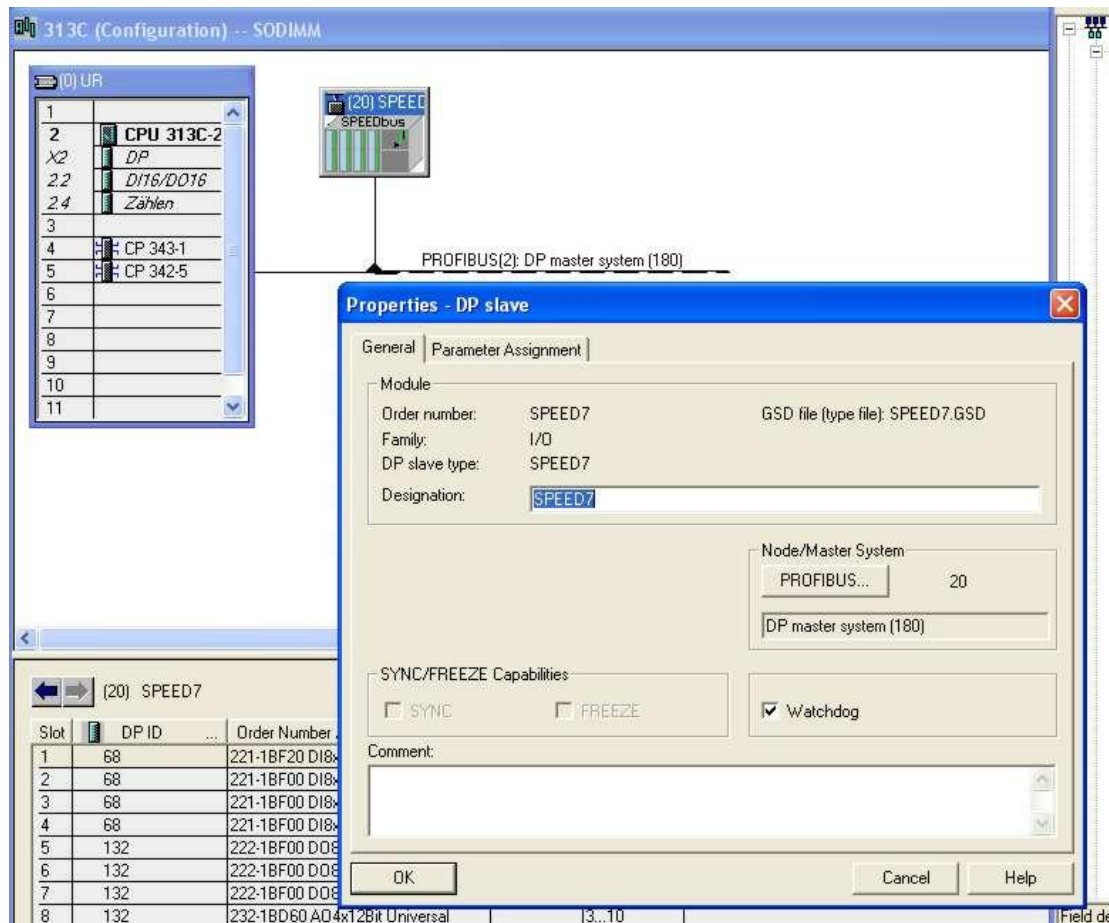
The Speed7 PLC 7001 is able to operate Speed7 200V modules. To configure this with the STEP7 HW Config you need a CP 342-5 Module added to your rack. This is found in **SIMANTIC 300 \ CP-300 \ PROFIBUS \ CP 342-5 \ 6GK7 342-5DA02-0XE0**. Access the configuration of this module and set [Operating Mode] DP-Master.

You can now attach a Speed7 module to this profibus which work as rack for Speed7 200V modules and is located in PROFIBUS DP → Additional Field Devices → I/O → SPEED7.

By dragging the  SPEED7 -symbol to the DP master system you get the properties monitor for this module. The Speed7 module must have profibus-address set to 20.

The Speed7 module supports up to 8 Speed7 200V





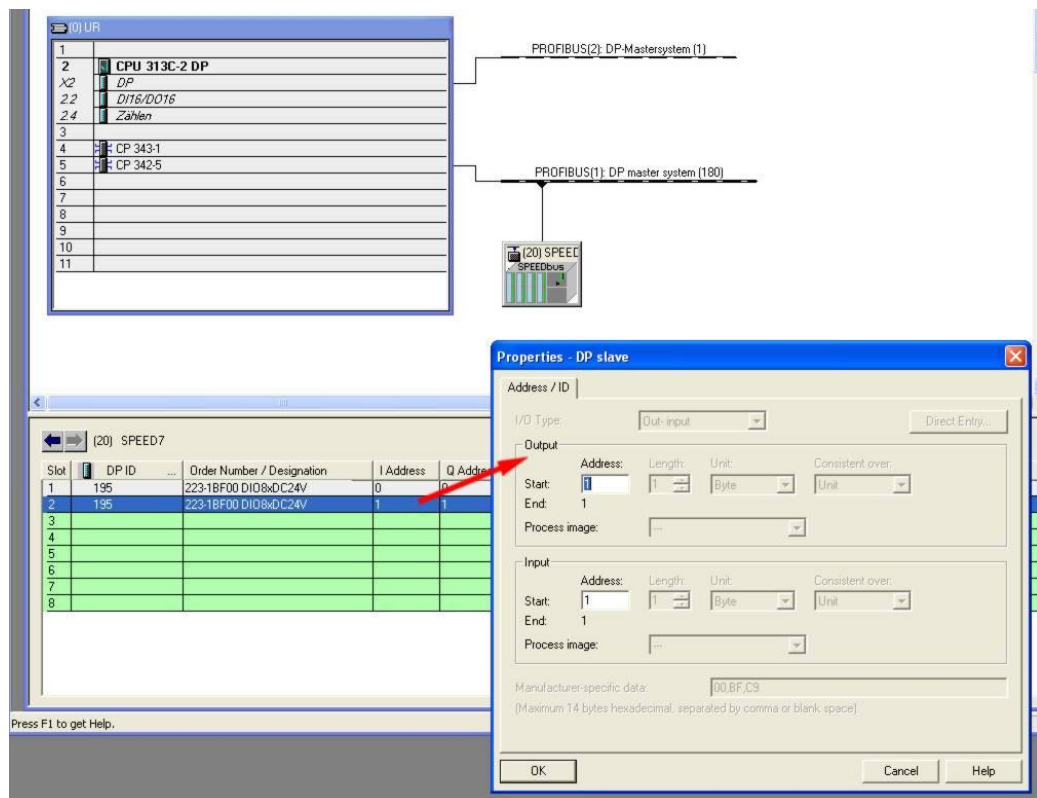
This is not a physically bus and only for configuration of the Speed7 200V modules.

Parameterization of modules

By using the SIMATIC Manager from Siemens you may set parameters for configurable System 200 modules at any time.

For this, double-click during the project engineering at the slot overview on the module you want to parameterize. Inside the appearing dialog window you may set the desired parameters.

Basic Operations



Parameterization during runtime

By using the SFCs 55, 56 and 57 you may alter and transfer parameters for selected modules during runtime.

For this you have to store the module specific parameters in so called "record sets".

More detailed information about the structure of the record sets is provided in the according module description.

4.7 Project transfer

Overview

There are the following possibilities for project transfer into the CPU:

- Transfer via MPI
- Transfer via MMC
- Transfer via integrated Ethernet PG/OP channel (Initialization necessary)

Transfer with MPI Programming cable

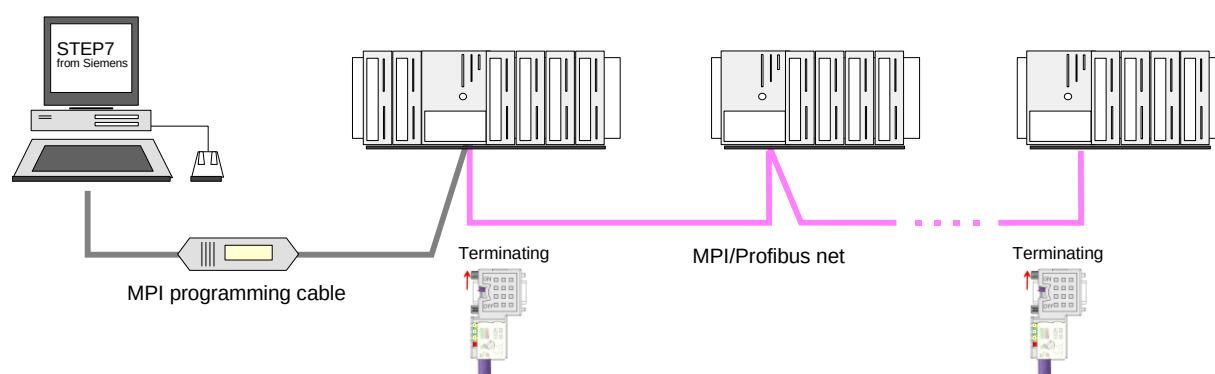
In order to program the CPU via MPI you will need a MPI programming cable which are available in different variants. The deployment of the cables is identical. The cables provide an RS485 connector for the MPI jack of the CPU and a RS232 or USB plug for the PC.

Due to the RS485 connection you may plug the MPI programming cables directly to an already plugged MPI connector on the MPI jack. Every bus participant identifies itself at the bus with a unique MPI address, in the course of which the address 0 is reserved for programming devices. The structure of a MPI net is widely identical with the structure of a 1.5MBaud Profibus net. I.e. the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and Profibus cables. Your PLC 7001 CPU supports transfer rates for MPI with 187.5kBaud. Default address is MPI address 2.

Terminating resistor

A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment.

Please make sure that the participants with the activated terminating resistors are always provided with voltage during start-up and operation.



Approach transfer via MPI

A maximum of 32 PG/OP connections is supported by MPI. The transfer via MPI takes place with the following proceeding:

- Connect your PC to the MPI jack of your CPU via a MPI programming cable.

Basic Operations

- Load your project in the SIMATIC Manager from Siemens.
- Choose in the menu **Options** > *Set PG/PC interface*.
- Select in the according list the "PC Adapter (MPI)"; if appropriate you have to add it first and then click on [Properties].
- Set in the register *MPI* the transfer parameters of your MPI net and type a valid *address*.
- Switch to the register *Local connection*.
- Set the COM port of the PCs and the transfer rate 38400Baud for the MPI programming cable from VIPA.
- Via **PLC** > *Load to module* you may transfer your project via MPI to the CPU and save it on a MMC via **PLC** > *Copy RAM to ROM* if one is plugged.

Transfer via MMC

As external storage medium a MMC is deployed.

The MMC (**M**ulti **M**edia **C**ard) serves as external transfer medium for programs and firmware for, among others, it provides the PC compatible FAT16 file system. With an overall reset or PowerON the MMC is automatically read. There may be stored several projects and sub-directories on a MMC storage module. Please take care that your current project engineering is stored in the root directory. For reading from the MMC the following file names must be used:

- Read after overall reset:
S7PROG.WLD (S7 project file)
PROTECT.WLD (Extended Know-how protection)
- Read after PowerON:
AUTOLOAD.WLD (S7 project file)

Transfer CPU → MMC

When the MMC has been installed, the write command stores the content of the battery buffered RAM as **S7PROG.WLD** at the MMC. The write command is controlled by means of the Siemens hardware configurator via **PLC** > *Copy RAM to ROM*. During the write process the "MCC"-LED of the CPU is blinking. When the LED expires, the write process is finished.

Process control

After a write process on the MMC, an according ID event is written into the diagnostic buffer of the CPU. To monitor the diagnosis entries, you select **PLC** > *Module Information* in the Siemens SIMATIC Manager. Via the register "Diagnostic Buffer" you reach the diagnosis window. At a successful write process the diagnostic buffer contains 0xE200.

When writing on the MMC, the following events may occur:

Event-ID	Meaning
0xE100	MMC access error
0xE101	MMC error file system
0xE102	MMC error FAT
0xE200	MMC writing finished

Transfer MMC → CPU

The transfer of the application program from the MMC into the CPU takes always place after an overall reset. The blinking of the LED "MCC" of the CPU marks the active transfer. An overall reset of the CPU takes place if the MMC does not contain a valid application program or if the transfer should fail. The red STOP-LED blinks three times.

Note!



If the size of the user application exceeds the user memory of the CPU, the content of the MMC is not transferred to the CPU.

Execute a compression before the transfer, for this does not happen automatically.

Transfer via Ether-net PG/OP channel (initialization necessary)

For the online access to the Ethernet PG/OP channel you have to assign IP address parameters by means of the "initialization".

After allocation the Ethernet PG/OP canal may be accessed by the IP address parameters.

Initialization

In the following the steps of initialization are described. More information may be found at "initialization" of the PG/OP channel.

- Determine the Ethernet (MAC) address of the Ethernet PG/OP channel. This always may be found as 1st address under the front flap of the CPU on a sticker on the left side.
- Establish a network connection between Ethernet PG/OP channel of the CPU and PC.
- Set at Siemens SIMATIC manager via **Options** > *Set PG/PC Interface* the access path to "TCP/IP -> Network card Protocol RFC 1006".
- Open with **PLC** > *Assign Ethernet Address* the dialog window for "initialization" of a station.
- Determine the CPU components via MAC address and assign it to IP address parameters. As long as the Ethernet PG/OP channel was not initialized yet, this owns the IP address 0.0.0.0 and the station name "Onboard PG/OP".

Transfer

Direct after the assignment the Ethernet PG/OP channel may be reached by the Siemens SIMATIC manager by means of these IP address parameters.

The transfer happens in the following approach:

- Open your project in the Siemens SIMATIC manager.
- Set at Siemens SIMATIC manager via **Options** > *Set PG/PC Interface* the access path to "TCP/IP -> Network card Protocol RFC 1006".
- Click to **PLC** > Download → the dialog "Select target module" is opened. Select your target module and enter the IP address parameters of the Ethernet PG/OP channel as address for connection. Provided that no new hardware configuration is transferred to the CPU, the given Ethernet-PG/OP channel is permanently stored in the project as transfer channel.

- With [OK] the transfer is started. System dependent you get a message that the projected system differs from target system. This message may be accepted by [OK] → your project is transferred and may be executed in the CPU after transfer.

4.8 Operating modes

Overview

The CPU can be in one of 4 operating modes:

- Operating mode STOP
- Operating mode START-UP
- Operating mode RUN
- Operating mode HOLD

Certain conditions in the operating modes START-UP and RUN require a specific reaction from the system program. In this case the application interface is often provided by a call to an organization block that was included specifically for this event.

Operating mode STOP

- The application program is not processed.
- If there has been a processing before, the values of counters, timers, flags and the process image are retained during the transition to the STOP mode.
- Outputs are inhibited, i.e. all digital outputs are disabled.
- RUN-LED off
- STOP-LED on

Operating mode START-UP

- During the transition from STOP to RUN a call is issued to the start-up organization block OB 100. The length of this OB is not limited. The processing time for this OB is not monitored. The START-UP OB may issue calls to other blocks.
- All digital outputs are disabled during the START-UP, i.e. outputs are inhibited.
- RUN-LED blinks
- STOP-LED off

When the CPU has completed the START-UP OB, it assumes the operating mode RUN.

Operating mode RUN

- The application program in OB 1 is processed in a cycle. Under the control of alarms other program sections can be included in the cycle.
- All timers and counters being started by the program are active and the process image is updated with every cycle.
- The BASP-signal (outputs inhibited) is deactivated, i.e. all digital outputs are enabled.
- RUN-LED on
- STOP-LED off

Operating mode HOLD

The CPU offers up to 4 breakpoints to be defined for program diagnosis. Setting and deletion of breakpoints happens in your programming environment. As soon as a breakpoint is reached, you may process your program step by step and in- and outputs can be activated.

Precondition

For the usage of breakpoints, the following preconditions have to be fulfilled:

- Testing in single step mode is only possible with STL. If necessary switch the view via **View > STL** to STL.
- The block must be opened online and must not be protected.
- The open block must not be altered in the editor.

Approach for working with breakpoints

- Activate **View > Breakpoint Bar**.
- Set the cursor to the command line where you want to insert a breakpoint.
- Set the breakpoint with **Debug > Set Breakpoint**. The according command line is marked with a circle.
- To activate the breakpoint click on **Debug > Breakpoints Active**. The circle is changed to a filled circle.
- Bring your CPU into RUN. When the program reaches the breakpoint, your CPU switches to the state HOLD, the breakpoint is marked with an arrow and the register contents are monitored.
- Now you may execute the program code step by step via **Debug > Execute Next Statement** or run the program until the next breakpoint via **Debug > Resume**.
- Delete (all) breakpoints with the option **Debug > Delete All Breakpoints**.

Behavior in operating state HOLD

- The LED RUN blinks and the LED STOP is on.
- The execution of the code is stopped. No level is further executed.
- All times are frozen.
- The real-time clock runs on.
- The outputs are closed, but may be released for test purposes.
- Passive CP communication is possible.



Note!

The usage of breakpoints is always possible. Switching to the operating mode test operation is not necessary.

With more than 3 breakpoints, a single step execution is not possible.

Basic Operations

Function security

The CPUs include security mechanisms like a Watchdog (100ms) and a cycle time surveillance which can be parameterized (min. 1ms) that stop res. execute a RESET at the CPU in case of an error and set it into a defined STOP state.

The PLC 7001 CPUs are developed function secure and have the following system properties:

Event	concerns	Effect
RUN → STOP	general	BASP (B efehls- A usgabe- S perre, i.e. command output lock) is set.
	central digital outputs	The outputs are set to 0V.
	central analog outputs	The voltage supply for the output channels is switched off.
	decentralized outputs	The outputs are set to 0V.
	decentralized inputs	The inputs are read constantly from the slave and the recent values are put at disposal.
STOP → RUN respectively PowerON	general	First the PII is deleted, and then OB 100 is called. After the execution of the OB, the BASP is reset and the cycle starts with: Delete PIQ → Read PII → OB 1.
	central analog outputs	The behavior of the outputs at restart can be preset.
	decentralized inputs	The inputs are read constantly from the slave and the recent values are put at disposal.
RUN	general	The program execution happens cyclically and can therefore be foreseen: Read PII → OB 1 → Write PIQ.

PII = Process image inputs

PIQ = Process image outputs

4.9 Overall reset

Overview

During the overall reset the entire user memory (RAM) is erased. Data located in the memory card is not affected.

You have 2 options to initiate an overall reset:

- initiate the overall reset by means of the function selector switch
- initiate the overall reset by means of the Siemens SIMATIC Manager



Note!

You should always issue an overall reset to your CPU before loading an application program into your CPU to ensure that all blocks have been cleared from the CPU.

Overall reset by means of the function selector

Condition

The operating mode of the CPU is STOP. Place the function selector on the CPU in position "STOP" → the STOP-LED is on.

Overall reset

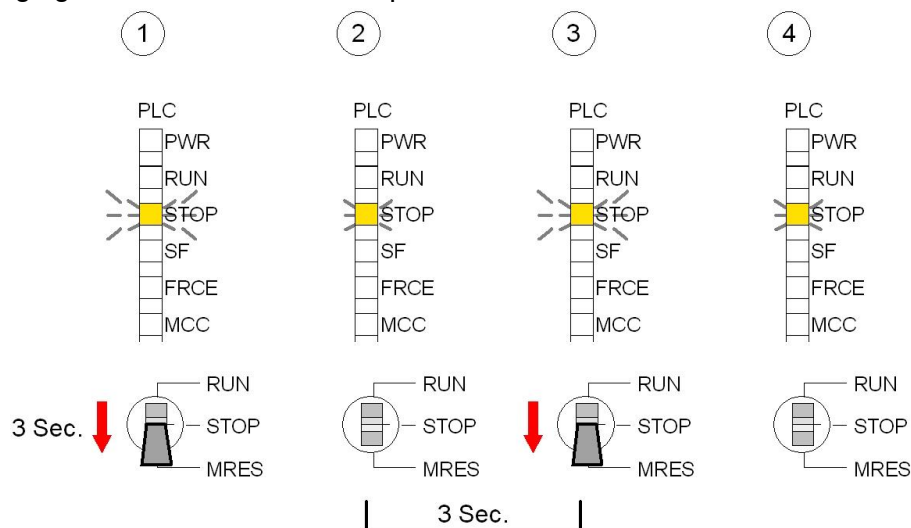
Place the function selector in the position MRES and hold it in this position for app. 3 seconds. → The STOP-LED changes from blinking to permanently on.

Place the function selector in the position STOP and switch it to MRES and quickly back to STOP within a period of less than 3 seconds.

→ The STOP-LED blinks (overall reset procedure).

The overall reset has been completed when the STOP-LED is on permanently. → The STOP-LED is on.

The following figure illustrates the above procedure:



Automatic reload

At this point the CPU attempts to reload the parameters and the program from the memory card. → The MCC-LED blinks.

When the reload has been completed the LED is turned off. The operating mode of the CPU will be STOP or RUN, depending on the position of the function selector.

Overall reset by means of the Siemens SIMATIC Manager

Condition

The operating mode of the CPU must be STOP.

You may place the CPU in STOP mode by the menu command **PLC > Operating mode**.

Overall reset

You may request the overall reset by means of the menu command **PLC > Clean/Reset**.

In the dialog window you may place your CPU in STOP mode and start the overall reset if this has not been done as yet.

The STOP-LED blinks during the overall reset procedure.

When the STOP-LED is on permanently the overall reset procedure has been completed.

Basic Operations

Automatic reload

At this point the CPU attempts to reload the parameters and the program from the memory card. → The MCC-LED blinks.

When the reload has been completed, the LED is turned off. The operating mode of the CPU will be STOP or RUN, depending on the position of the function selector.

Set back to factory setting

The following approach deletes the internal RAM of the CPU completely and sets it back to the delivery state.

Please regard that the MPI address is also set back to default 2!

- Push down the reset lever for app. 30 seconds. The ST-LED blinks. After a few seconds the LED turns to static light. Count the number of static light phases because now the LED switches between static light and blinking.
- After the 6th static light you release the reset lever and push it down again shortly. Now the green RUN-LED is on once. This means that the RAM is totally deleted.
- Turn the power supply off and on again.

More information may be found at the part "Factory reset" further below.

4.10 Firmware update

Overview

By means of an MMC there is the opportunity to execute a firmware update at the CPU. For this an accordingly prepared MMC must be in the CPU during the start-up.

Latest Firmware

For latest firmware version mail to support@profichip.com or visit <http://www.profichip.com/products/speed7-technology/single-board-plc/>



Attention!

When installing a new firmware you have to be extremely careful. Under certain circumstances you may destroy the CPU, for example if the voltage supply is interrupted during transfer or if the firmware file is defective.

In this case, please call the profichip-support!

Please make sure that the version of the update firmware is different to the existing firmware otherwise no update is executed.

Display the Firmware version of the SPEED7 system via web page

The PLC 7001 CPU has an integrated web page that monitors information about firmware version of the I/O components. The Ethernet PG/OP channel provides the access to this web page.

To activate the PG/OP channel you have to enter according IP parameters.

This can be made in Siemens SIMATIC manager either by a hardware configuration, loaded by MMC respectively MPI or via Ethernet by means of the MAC address with **PLC > Assign Ethernet Address**.

After that you may access the PG/OP channel with a web browser via the IP address of the project engineering. More detailed information is to find in "Access to the internal web page".

Determine CPU firmware version with module information

First establish an online connection to the CPU. To monitor the module information you choose the option **PLC > Module Information** in the Siemens SIMATIC Manager.

Via the register "General" the window with hardware and firmware version may be selected. From software-technical reasons there is something different of the CPU 313-6CF03 to the CPU 313C - 2 DB from Siemens:

The releases of hard and software may be found at "Order No./Description". Here the number at "Version" is irrelevant.

Description:

CPU 313C-2 DP

System Identification:

SIMATIC 300

Name:

CPU 313C-2 DP

Version:

Order No. / Description	Component	Version
6ES7 313-6CF03	Hardware	1
7001 AG01.10 V3.4.1	Firmware	V 2.0.0

Hardware:

Release

Sub version

Firmware:

Version

irrelevant



Note!

Every register of the module information dialog is supported by the PLC 7001 CPUs. More about these registers may be found in the online help of the Siemens SIMATIC manager.

Load firmware and transfer it to MMC

- Get the latest firmware from support@profichip.com
- Extract the zip-file and copy the extracted file to your MMC.
- Following this approach, transfer all wanted firmware files to your MMC.



Attention!

With a firmware update an overall reset is executed automatically. If your program is only available in the load memory of the CPU it is deleted! Save your program before executing a firmware update! After the firmware update you should execute a "Set back to factory settings" (see following page).

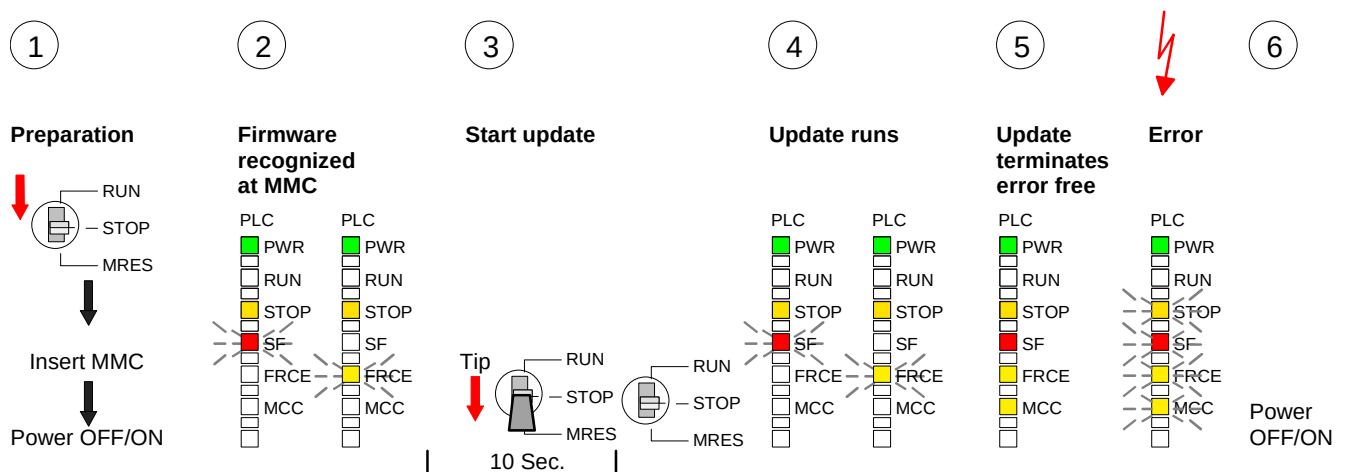
Basic Operations

Transfer firmware from MMC into CPU

1. Get the RUN-STOP lever of your CPU in position STOP. Turn off the voltage supply. Plug the MMC with the firmware files into the CPU. Please take care of the correct orientation of the MMC. Turn on the voltage supply.
2. After a short boot-up time, the alternate blinking of the LEDs SF and FRCE shows that at least one newer firmware file was found on the MMC.
3. You start the transfer of the firmware as soon as you tip the RUN/STOP lever downwards to MRES within 10s.
4. During the update process, the LEDs SF and FRCE are blinking alternately and the MMC LED is on. This may last several minutes.
5. The update is finished successfully when the LEDs PWR, STOP, SF, FRCE and MCC are on. If they are blinking fast, an error occurred.
6. Turn Power OFF and ON. Now it is checked by the CPU whether newer firmware versions are available on the MMC. If so, again the LEDs SF and FRCE flash after a short start-up period. Continue with step 3.

If the LEDs do not flash, the firmware update is completed.

Now a *factory reset* should be executed (see below). After that the CPU is ready for duty.



4.11 Factory reset

Proceeding

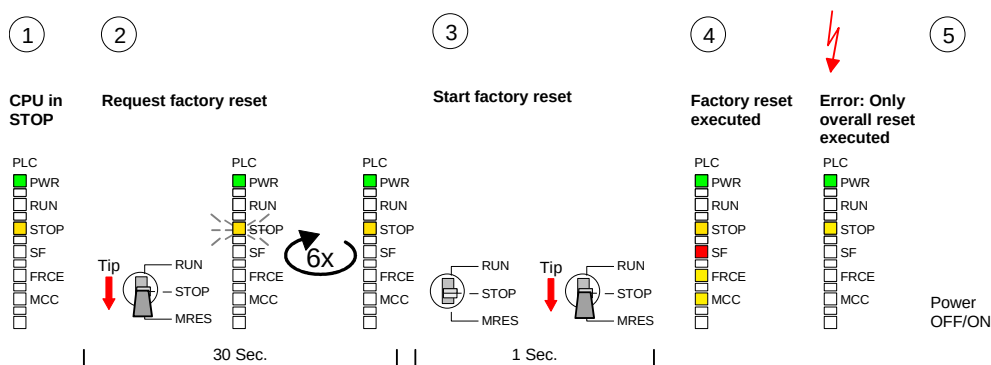
With the following proceeding the internal RAM of the CPU is completely deleted and the CPU is reset to delivery state.

Please note that here also the IP address of the Ethernet PG/OP channel is set to 0.0.0.0 and the MPI address is reset to the address 2!

A factory reset may also be executed by the MMC-Cmd `FACTORY_ RESET`. More information may be found at "MMC-Cmd - Auto commands".

1. Switch the CPU to STOP.
2. Push the operating switch down to position MRES for 30s. Here the STOP-LED flashes. After a few seconds the stop LED changes to static light. Now the STOP LED changes between static light and flashing. Starting here count the static light states.
3. After the 6th static light release the operating mode switch and tip it downwards to MRES within 1s.
4. For the confirmation of the resetting procedure the LEDs PWR, STOP, SF, FRCE and MCC get ON. If not, the factory reset has failed and only an overall reset was executed. In this case you can repeat the procedure. A factory reset can only be executed if the stop LED has static light for exactly 6 times.
5. After factory reset switch the power supply off and on.

The proceeding is shown in the following Illustration:



Note!

After the firmware update you always should execute a *Factory reset*.

4.12 Memory extension with MCC



Overview

The memory of the PLC 7001 CPU may be extended from 64kB up to 512kB by using a memory extension card (MCC) from VIPA.

The MCC is a specially prepared MMC (**M**ultimedia **C**ard). When this card is available to the CPU (connecting it to the according pins with an MMC slot) the memory will be extended after an overall reset. Only one memory expansion may be activated at the same time.

On the MCC there is the file *memory.key*. This file may not be altered or deleted. You may use the MCC also as "normal" MMC for storing your project.

Approach

To extend the memory, make the MCC available to the CPU via a MMC slot and perform an overall reset.

If the memory expansion on the MCC exceeds the maximum extendable memory range of the CPU, the maximum possible memory of the CPU is automatically used.

You may determine the recent memory extension via the Siemens SIMATIC Manager at *Module Information* - "Memory".



Attention!

Please make sure that the MCC remains plugged when you've executed the memory expansion at the CPU. Otherwise the CPU switches to STOP after 48h. The MCC can not be exchanged with a MCC of the same memory configuration.

Behavior

When the MCC memory configuration has been taken over you may find the diagnosis entry 0xE400 in the diagnostic buffer of the CPU.

After pulling the MCC the entry 0xE401 appears in the diagnostic buffer, the SF-LED is on and after 72h the CPU switches to STOP. A reboot is only possible after plugging-in the MCC again or after an overall reset.

After re-plugging the MCC, the SF-LED extinguishes and 0xE400 is entered into the diagnostic buffer.

You may reset the memory configuration of your CPU to the initial status at any time by executing an overall reset without MCC.

4.13 Extended know-how protection

Overview

Besides the "standard" Know-how protection the SODIMM PLC 7001 from Profichip provides an "extended" know-how protection that serves a secure block protection for accesses of 3rd persons.

Standard protection

The standard protection from Siemens transfers also protected blocks to the PG but their content is not displayed. But with according manipulation the Know-how protection is not guaranteed.

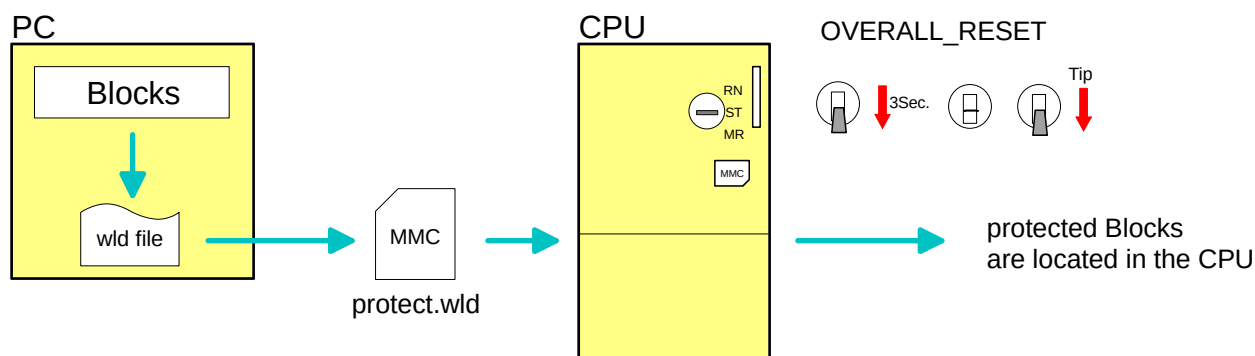
Extended protection

The "extended" know-how protection offers the opportunity to store blocks permanently in the CPU.

At the "extended" protection you transfer the protected blocks into a WLD-file named protect.wld. By plugging the MMC and following overall reset, the blocks in the protect.wld are permanently stored in the CPU.

You may protect OBs, FBs and FCs.

When back-reading the protected blocks into the PG, exclusively the block header are loaded. The source remains in the CPU and is thus protected for accesses of 3rd persons.



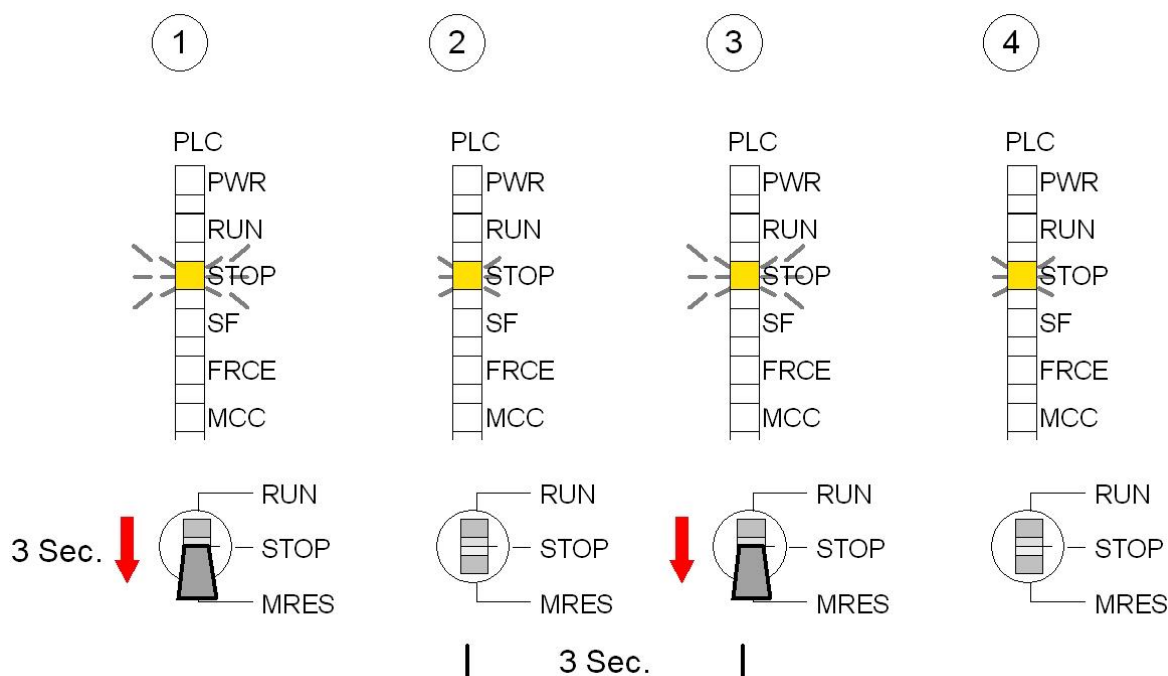
Protect blocks with protect.wld

Create a new wld-file in your project engineering tool with **File > Memory Card file > New** and rename it to "protect.wld".

Transfer the according blocks into the file by dragging them with the mouse from the project to the file window of protect.wld.

Transfer protect.wld to CPU with overall reset

Transfer the file protect.wld to a MMC storage module, plug the MMC into the CPU and execute an overall reset with the following approach:



The overall reset stores the blocks in protect.wld permanently in the CPU protected from accesses of third persons.

Protection behavior

Protected blocks are overwritten by a new protect.wld.

Using a PG 3rd persons may access protected blocks but only the block header is transferred to the PG. The protected block code remains in the CPU and can not be read.

Changing or deleting protected blocks

Protected blocks of the CPU may be substituted at any time by blocks with the same name in the RAM. This change remains up to next overall reset. Protected blocks may only be permanently overwritten if these are deleted at the protect.wld before.

By transferring an empty protect.wld from the MMC you may delete all protected blocks in the CPU.

Usage of protected blocks

Due to the fact that reading of a "protected" block from the CPU monitors no symbol labels it is convenient to provide the "block covers" for the end user.

For this, create a project out of all protected blocks. Delete all networks in the blocks so that these only contain the variable definitions in the according symbolism.

4.14 MMC-Cmd - Auto commands

Overview

A *command file* at a MMC may be started automatically when the MMC is plugged and the CPU is in STOP. As soon as the MMC is stuck the command file is once executed at CPU STOP up to the next PowerON.

The command file is a text file, which consists of a command sequence to be stored as ***vipa_cmd.mmc*** in the root directory of the MMC.

The file has to be started by *CMD_START* as 1st command, followed by the desired commands (no other text) und must be finished by *CMD_END* as last command.

Text after the last command *CMD_END* e.g. comments is permissible, because this is ignored. As soon as the command file is recognized and executed each action is stored at the MMC in the log file logfile.txt. In addition for each executed command a diagnostics entry may be found in the diagnostics buffer.

Commands

In the following there is an overview of the commands. Please regard the command sequence is to be started with *CMD_START* and ended with *CMD_END*.

Command	Description	Diagnostics entry
CMD_START	In the first line <i>CMD_START</i> is to be located.	0xE801
	There is a diagnostic entry if <i>CMD_START</i> is missing	0xE8FE
WAIT1SECOND	Waits ca. 1 second.	0xE803
WEBPAGE	The current web page of the CPU is stored at the MMC as "webpage.htm".	0xE804
LOAD_PROJECT	The function "Overall reset and reload from MMC" is executed. The wld file located after the command is loaded else "s7prog.wld" is loaded.	0xE805
SAVE_PROJECT	The recent project (blocks and hardware configuration) is stored as "s7prog.wld" at the MMC. If the file just exists it is renamed to "s7prog.old".	0xE806
FACTORY_RESET	Executes "factory reset".	0xE807
DIAGBUF	The current diagnostics buffer of the CPU is stored as "diagbuff.txt" at the MMC.	0xE80B
SET_NETWORK	IP parameters for Ethernet PG/OP channel may be set by means of this command. The IP parameters are to be given in the order IP address, subnet mask and gateway in the format xxx.xxx.xxx.xxx each separated by a comma. Enter the IP address if there is no gateway used.	0xE80E
CMD_END	In the last line <i>CMD_END</i> is to be located.	0xE802

Examples

Basic Operations

The structure of a command file is shown in the following. The corresponding diagnostics entry is put in parentheses.

Example 1

CMD_START	Marks the start of the command sequence (0xE801)
LOAD_PROJECT proj.wld	Execute an overall reset and load "proj.wld" (0xE805)
WAIT1SECOND	Wait ca. 1s (0xE803)
WEBPAGE	Store web page as "webpage.htm" (0xE804)
DIAGBUF	Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)
CMD_END	Marks the end of the command sequence (0xE802)
... arbitrary text ...	Text after the command CMD_END is not evaluated.

Example 2

CMD_START	Marks the start of the command sequence (0xE801)
LOAD_PROJECT proj2.wld	Execute an overall reset and load "proj2.wld" (0xE805)
WAIT1SECOND	Wait ca. 1s (0xE803)
WAIT1SECOND	Wait ca. 1s (0xE803)
SET_NETWORK 172.16.129.210,255.255.224.0,172.16.129.210	IP parameter (0xE80E)
WAIT1SECOND	Wait ca. 1s (0xE803)
WAIT1SECOND	Wait ca. 1s (0xE803)
WEBPAGE	Store web page as "webpage.htm" (0xE804)
DIAGBUF	Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)
CMD_END	Marks the end of the command sequence (0xE802)
... arbitrary text ...	Text after the command CMD_END is not evaluated.



Note!

The parameters IP address, subnet mask and gateway may be received from the system administrator.
Enter the IP address if there is no gateway used.

4.15 CPU specific diagnostic entries

Entries in the diagnostic buffer

You may read the diagnostic buffer of the PLC 7001 via the Siemens SIMATIC Manager. Besides of the standard entries in the diagnostic buffer, the PLC 7001 CPUs support some additional CPU specific entries in form of event-IDs.

The current content of the diagnostics buffer is stored on MMC by means of the MMC-Cmd DIAGBUF. More information may be found at "MMC-Cmd - Auto commands".

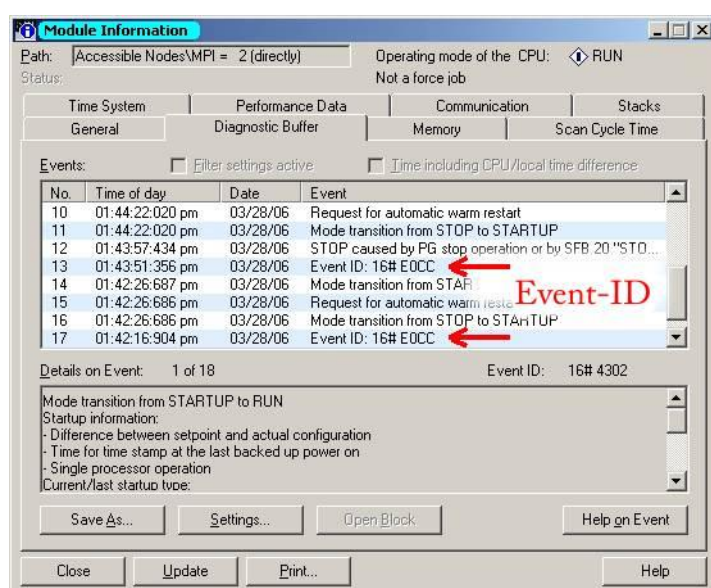


Note!

Every register of the module information is supported by the PLC 7001 CPU. More information may be found at the online help of the Siemens SIMATIC manager.

Monitoring the diagnostic entries

To monitor the diagnostic entries you choose the option **PLC > Module Information** in the Siemens SIMATIC Manager. Via the register "Diagnostic Buffer" you reach the diagnostic window:



The diagnosis is independent from the operating mode of the CPU. You may store a maximum of 100 diagnostic entries in the CPU.

The following page shows an overview of the CPU specific Event-IDs.

Overview of the Event-IDs

Event-ID	Description
0xE003	Error at access to I/O devices Zinfo1: I/O address Zinfo2: Slot
0xE004	Multiple parameterization of a I/O address Zinfo1: I/O address Zinfo2: Slot
0xE005	Internal error – Please contact the Profichip-support!
0xE006	Internal error – Please contact the Profichip-support!
0xE007	Configured in-/output bytes do not fit into I/O area
0xE008	Internal error – Please contact the Profichip-support!
0xE009	Error at access to standard back plane bus
0xE010	Not defined module group at backplane bus recognized Zinfo2: Slot

Basic Operations

Event-ID	Description
	Zinfo3: Type ID
0xE011	Master project engineering at Slave-CPU not possible or wrong slave configuration
0xE012	Error at parameterization
0xE013	Error at shift register access to VBUS digital modules
0xE014	Error at Check_Sys
0xE015	Error at access to the master Zinfo2: Slot of the master (32=page frame master)
0xE016	Maximum block size at master transfer exceeded Zinfo1: I/O address Zinfo2: Slot
0xE017	Error at access to integrated slave
0xE018	Error at mapping of the master I/O devices
0xE019	Error at standard back plane bus system recognition
0xE01A	Error at recognition of the operating mode (8 / 9 Bit)
0xE01B	Error - maximum number of plug-in modules exceeded
0xE0B0	Speed7 is not stoppable (probably undefined BCD value at timer)
0xE0C0	Not enough space in work memory for storing code block (block size exceeded)
0xE0CC	Communication error MPI / Serial
0xE100	MMC access error
0xE101	MMC error file system
0xE102	MMC error FAT
0xE104	MMC error at saving
0xE200	MMC writing finished (Copy Ram2Rom)
0xE210	MMC reading finished (reload after overall reset)
0xE400	Memory expansion MCC has been plugged
0xE401	Memory expansion MCC has been removed
0xE801	MMC-Cmd: CMD_START recognized and successfully executed
0xE802	MMC-Cmd: CMD_END recognized and successfully executed
0xE803	MMC-Cmd: WAIT1SECOND recognized and successfully executed
0xE804	MMC-Cmd: WEBPAGE recognized and successfully executed
0xE805	MMC-Cmd: LOAD_PROJECT recognized and successfully executed
0xE806	MMC-Cmd: SAVE_PROJECT recognized and successfully executed
0xE807	MMC-Cmd: FACTORY_RESET recognized and successfully executed
0xE80B	MMC-Cmd: DIAGBUF recognized and successfully executed

Event-ID	Description
0xE80E	MMC-Cmd: SET_NETWORK recognized and successfully executed
0xE8FB	MMC-Cmd: Error: Initialization of the Ethernet PG/OP channel by means of SET_NETWORK is faulty.
0xE8FC	MMC-Cmd: Error: Not every IP-Parameter is set at SET_NETWORK.
0xE8FE	MMC-Cmd: Error: CMD_START was not found
0xE8FF	MMC-Cmd: Error: Reading the CMD file is faulty (MMC error)
0xE901	Check sum error
0xEA00	Internal error – Please contact the Profichip-support!
0xEA01	Internal error – Please contact the Profichip-support!
0xEA04	SBUS: Multiple parameterization of a I/O address Zinfo1: I/O address Zinfo2: Slot Zinfo3: Data width
0xEA05	Internal error – Please contact the Profichip-support!
0xEA07	Internal error – Please contact the Profichip-support!
0xEA08	SBUS: Parameterized input data width unequal to plugged input data width Zinfo1: Parameterized input data width Zinfo2: Slot Zinfo3: Input data width of the plugged module
0xEA09	SBUS: Parameterized output data width unequal to plugged output data width Zinfo1: Parameterized output data width Zinfo2: Slot Zinfo3: Output data width of the plugged module
0xEA10	SBUS: Input address outside input area Zinfo1: I/O address Zinfo2: Slot Zinfo3: Data width
0xEA11	SBUS: Output address outside output area Zinfo1: I/O address Zinfo2: Slot Zinfo3: Data width
0xEA12	SBUS: Error at writing record set Zinfo1: Slot Zinfo2: Record set number Zinfo3: Record set length
0xEA14	SBUS: Multiple parameterization of a I/O address (Diagnostic address) Zinfo1: I/O address Zinfo2: Slot

Event-ID	Description
	Zinfo3: Data width
0xEA15	Internal error - Please contact the Profichip-support!
0xEA18	SBUS: Error at mapping of the master I/O devices Zinfo2: Master slot
0xEA19	Internal error - Please contact the Profichip-support!
0xEA20	Error - RS485 interface is not set to Profibus DP master but there is a Profibus DP master configured.
0xEA21	Error - Project engineering RS485 interface X2/X3: Profibus DP master is configured but missing Zinfo2: Interface x
0xEA22	Error - RS485 interface X2 - value is out of range Zinfo: Configured value X2
0xEA23	Error - RS485 interface X3 - value is out of range Zinfo: Configured value X3
0xEA24	Error - Project engineering RS485 interface X2/X3: Interface/Protocol is missing, the default settings are used. Zinfo2: Configured value X2 Zinfo2: Configured value X3
0xEA30	Internal error – Please contact the Profichip-support!
0xEA98	Timeout at waiting for reboot of a SBUS module (Server)
0xEA99	Error at file reading via SBUS
0xEE00	Internal error - Please contact the Profichip-support!

4.16 Using test functions for control and monitoring of variables

Overview

For troubleshooting purposes and to display the status of certain variables you can access certain test functions via the menu item **Debug** of the Siemens SIMATIC Manager.

The status of the operands and the VKE can be displayed by means of the test function **Debug > Monitor**.

You can modify and/or display the status of variables by means of the test function **PLC > Monitor/Modify Variables**.

Debug > Monitor

This test function displays the current status and the VKE of the different operands while the program is being executed.

It is also possible to enter corrections to the program.



Note!

When using the test function “Monitor” the PLC must be in RUN mode!

The processing of statuses can be interrupted by means of jump commands or by timer and process-related alarms. At the breakpoint the CPU stops collecting data for the status display and instead of the required data it only provides the PG with data containing the value 0. For this reason, jumps or time and process alarms can result in the value displayed during program execution remaining at 0 for the items below:

- the result of the logical operation VKE
- Status / AKKU 1
- AKKU 2
- Condition byte
- absolute memory address SAZ. In this case SAZ is followed by a "?".

The interruption of the processing of statuses does not change the execution of the program. It only shows that the data displayed is no longer valid

PLC > Monitor/Modify Variables

This test function returns the condition of a selected operand (inputs, outputs, flags, data word, counters or timers) at the end of program-execution.

This information is obtained from the process image of the selected operands. During the "processing check" or in operating mode STOP the periphery is read directly from the inputs. Otherwise only the process image of the selected operands is displayed.

Control of outputs

It is possible to check the wiring and proper operation of output-modules.

You can set outputs to any desired status with or without a control program. The process image is not modified but outputs are no longer inhibited.

Control of variables

The following variables may be modified:

E, A, M, T, Z and D.

The process image of binary and digital operands is modified independently of the operating mode of the PLC 7001 CPU.

When the operating mode is RUN the program is executed with the modified process variable. When the program continues they may, however, be modified again without notification.

Process variables are controlled asynchronously to the execution sequence of the program.

5 I/O periphery

Overview

This chapter contains all information necessary for the employment of the in-/output periphery of the CPU 313-6CF03. It describes functionality, project engineering and diagnostic of the digital part.

5.1 Overview

General

The SPEED7 PLC 7001 module is an embedded system.

Project engineering

The project engineering takes place in the Siemens SIMATIC manager as CPU 313C-2DP from Siemens (6ES7 313-6CF03-0AB0 V2.0).

Here the CPU is parameterized by the "Properties" dialog of the CPU 313C-2DP.

For parameterization of the digital I/O periphery and the technological functions the corresponding sub module of the CPU 313C-2DP may be used.

By including the SPEED7.gsd the specific parameters of the PLC 7001 CPU may be set during hardware configuration. So for example the integrated RS485 interface may be parameterized.

I/O periphery

The integrated I/Os of the PLC 7001 may be used for technological functions or as standard I/Os. Technological functions and standard I/Os may be used simultaneously with appropriate hardware. Read access to inputs used by technological functions is possible. Write access to used outputs is not possible.

Technological functions

Up to 3 channels may be parameterized as technological function. The parameterization of the appropriate channel is made in the hardware configurator by the *count* sub module of the CPU 313C-2DP.

There are the following technological functions:

- Continuous count
- Single count
- Periodic count

The controlling of the corresponding counter mode happens by means of the SFB COUNT (SFB 47) of the user program.

5.2 In-/Output range

Overview CPU 313-6CF03

The PLC 7001 has the following digital in- and output ranges integrated in this module:

- Digital Input: 16xDC 3,3V
- Digital Output: 16xDC 3,3V
- Technological functions: 3 Channels

Via the parameterization you may assign up to 16 on-chip alarm functionality which results in low latency and fast response times upon critical system conditions und provides the capability to capture time critical events very accurately. Additionally 16 of the digital inputs can be parameterized as counter.

5.3 Digital part

Digital part

The digital part consists of 16 input, 16 output and 3 channels for technological functions. By means of the parameterization you may assign interrupt properties to the inputs I+0.0 to I+1.7.

Access to the digital part

The CPU 313-6CF03 creates in its peripheral area an area for input respectively output data. Without a hardware configuration the in the following specified default addresses are used.

Input range

Sub module	Default address	Access	Assignment
<i>DI16/DO16</i>	124	Byte	Digital Input DI00 ... DI07
	125	Byte	Digital Input DI08 ... DI15
<i>Count</i>	768	DInt	Channel 0: Count value / Frequency value
	772	DInt	Channel 1: Count value / Frequency value
	776	DInt	Channel 2: Count value / Frequency value
	780	DInt	reserved

Output range

Sub module	Default address	Access	Assignment
<i>DI16/DO16</i>	124	Byte	Digital Output DO00 ... DO07
	125	Byte	Digital Output DO08 ... DO15
<i>Count</i>	768	DWord	reserved
	772	DWord	reserved
	776	DWord	reserved
	780	DWord	reserved

5.4 Digital part – Parameterization**Parameter data**

Parameters of the digital part may be set by means of the *DI16/DO16* sub module of the CPU 313C-2DP from Siemens during hardware configuration.

In the following all parameters are specified, which may be used with the hardware configuration of the digital periphery.

General

This provides the short description of the digital periphery. At *Comment* information about the module such as purpose may be entered.

Addresses

At this register the start address of the in-/output periphery may be set.

Inputs

Here there are the following adjustment possibilities:

- Hardware interrupt
- Input delay

For the digital output channels there are no parameters.

Hardware interrupt

A hardware interrupt may be optionally triggered on the rising or falling edge of an input. A diagnostic interrupt is only supported together with hardware interrupt lost.

Select with the arrow keys the input and activate the desired hardware interrupt.

5.5 Counter - Brief Introduction

Overview

The CPU 313-6CF03 has in-/outputs, which may be used for technological functions respectively as standard periphery. Technological functions and standard I/O may be used simultaneously with appropriate hardware.

Read access to inputs used by technological functions is possible. Write access to used outputs is not possible.

The parameterization of the corresponding channel is made in the hardware configurator by means of the *Count* sub module of the CPU 312C from Siemens.

Now the following technological functions at 2 channels are at the disposal:

- Continuous count, e.g. for position decoding with Incremental encoder
- Single count, e.g. for unit decoding to a maximum limit
- Periodical count, e.g. for applications with repeated counting operations

Independent of the number of activated counters for CPU 313-6CF03 the maximum frequency amounts to 10kHz.

The controlling of the appropriate modes of operation is made from the user program by the SFB COUNT (SFB 47).

Preset respectively parameterize counter

The counter signal is detected and evaluated during counting operation. Every counter occupies one double word in the input range for the *counter register*. In the operating modes "single count" and "periodical count" an end respectively start value may be defined according to the counting direction up respectively down.

Each counter can be parameterized with additional functions such as gate function, latch function, comparison value, hysteresis and hardware interrupt.

Each counter parameter may be set by the *Count* sub module of the Siemens CPU 313C-2DP. Here is defined among others:

- Interrupt behavior
- Max. Frequency
- Counter mode respectively behavior
- Stat, end, comparison value and hysteresis

Controlling the counter functions

The SFB COUNT (SFB 47) should cyclically be called (e.g. OB 1) for controlling the counter functions. The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.

Among others the SFB 47 contains a request interface. Hereby you get read and write access to the registers of the appropriate counter.

So that a new job may be executed, the previous job must have be finished with JOB_DONE = TRUE. Per channel you may call the SFB in each case with the same instance DB, since the data necessary for the internal operational are stored here. Writing accesses to outputs of the instance DB is not permissible.

**Note!**

You must not call an SFB you have configured in your program in another program section under another priority class, because the SFB must not interrupt itself.

Example: It is not allowed to call the same SFB both in OB 1 and in the interrupt OB.

Controlling the counter

The counter is controlled by the internal gate (i gate). The i gate is the result of logic operation of hardware gate (HW gate) and software gate (SW gate), where the HW gate evaluation may be deactivated by the parameterization.

HW gate:	open (activate):	Edge 0-1 at hardware gate _x input of the module
	close (deactivate):	Edge 1-0 at hardware gate _x input of the module
SW gate:	open (activate):	In application program by setting SW_GATE of the SFB 47
	close (deactivate):	In application program by resetting SW_GATE of the SFB 47

Read counter

The counter values may be read by the output parameter COUNTVAL of the SFB 47. There is also the possibility for direct access to the counter values by means of the input address of the *Count* sub module.

Counter inputs (Connections)

There are the following possibilities for connection to the technological functions:

- 3.3V incremental encoder, equipped with two tracks with 90° phase offset
- 3.3V pulse generator with direction signal
- 3.3V proximity switch (e.g. BERO or light barrier)

For not all inputs are available at the same time, you may set the input assignment for every counter via the parameterization. For each counter the following inputs are available:

Channel_x (A)

Pulse input for count signal res. track A of an encoder. Here you may connect encoder with 1-, 2- or 4-tier evaluation.

Channel_x (B)

Direction signal res. track B of the encoder. Via the parameterization you may invert the direction signal.

Hardware gate_x

This input allows you to open the HW gate with a high peek and thus start a count process. The usage of the HW gate may be parameterized.

Latch_x

With an edge 0-1 at Latch_x the recent counter value is stored in a memory that you may read at need.

Counter outputs

Every counter has an assigned output channel. The following behavior for the output channel may be set via parameterization:

- No comparison: Output is not controlled and is switched in the same way as a normal output.
- Count value $\geq$ comparison value:
Output is set as long as counter value $\geq$ comparison value.

- Count value $\leq$ comparison value:
Output is set as long as counter value $\leq$ comparison value.
- Pulse at comparison value: You can specify a pulse period for adaptation to the actuators you are using. The output is set for the given pulse duration, as soon as the counter reached the comparison value. If you have parameterized a main count direction the output is only set when reaching the comparison value from the main counting direction. The maximum pulse duration may amount to 510ms. By setting 0 as pulse duration the output gets set as long as the comparison conditions are fulfilled.

Parameter overview

In the following the parameters are listed which may be used for counter configuration during hardware configuration.

General

Here the short description of the counter function may be found. At Comment information about the module such as purpose may be entered.

Addresses

Here the start address of the in- output periphery is set.

Basic parameters

Here the interrupts the counter functions should trigger may be selected. You have the following options:

- None: No interrupt is triggered.
- Process: The counting function triggers a hardware interrupt.
- Diagnostics and Process: With the CPU 312SC the diagnostic interrupt of the digital in-/output periphery is only supported in connection with "hardware interrupt lost".

Count

Parameters	Description	Range of values	Default
Main count direction	• <i>None</i>: No restriction of the counting range • <i>Up</i>: Restricts the up-counting range. Counter starts at 0 or load value, counts in positive direction up to the declaration end value -1 and then jumps back to load value at the next positive transducer pulse. • <i>Down</i>: Restricts the down-counting range. The Counter starts at the declared start value or load value in negative direction, counts to 1 and then jumps to start value at the next negative encoder pulse.	• None • Up • Down (not with continuous count)	None
End value/ Start value	<i>End value</i> , with up-count as default. <i>Start value</i> , with down-count as default.	2...2147483647 ($2^{31}-1$)	2147483647 ($2^{31}-1$)
Gate function	• <i>Cancel count</i>: The count starts when the gate opens and resumes at the load value when the gate opens again. • <i>Stop count</i>: The count is interrupted when the gate closes and resumed at the last actual value when the gate opens again.	• Abort the count operation • Interrupt the count operation	Cancel count
Comparison value	The count value is compared with the comparison value. see also the parameter "Characteristics of the output": • No main direction of count • Up-count as default • Down-count as default	-2^{31} to $+2^{31}-1$ -2^{31} to End value-1 1 to $+2^{31}-1$	0
Hysteresis	A hysteresis is used to eliminate frequent output jitter if the count value lies within the range of the comparison value. 0 and 1 means: Hysteresis switched off	0 to 255	0
max. frequency: counting signals/hard- ware gate	You can set the maximum frequency of the track A/pulse, track B/direction and hardware gate signals in fixed steps.	10, 5, 2, 1kHz	10kHz
max. frequency: Latch	You can set the maximum frequency of the latch signal in fixed steps.	10, 5, 2, 1kHz	10kHz
Signal evaluation	The count and direction signals are connected to the input. A rotary transducer is connected to the input (single, dual or quadruple evaluation).	• Pulse/Direction • Rotary encoder single • Rotary encoder, double • Rotary encoder quadruple	Pulse/Direction
Hardware gate	In the activated state the Gate control is made via SW-gate and HW-gate, otherwise via SW-gate only.	• activated • deactivated	deactivated

Parameters	Description	Range of values	Default
Count direction inverted	In the activated state the "direction" input signal is inverted.	• activated • deactivated	deactivated
Characteristic of the output	The output and the "Comparator" (STS_CMP) status bit are set, dependent on this parameter.	• No comparison • Count $\geq$ comparison value • Count $\leq$ comparison value • Pulse at comparison value	No comparison
Pulse duration	With the setting "Characteristics of the output: Pulse at comparison value" the pulse duration of the output signal may be specified. Only even values are possible. The value is internal multiplied with 1.024ms.	0 to 510	0
Hardware interrupt: Hardware gate opening	In the activated state a hardware interrupt is generated when the hardware gate opens while the software gate is open.	• activated • deactivated	
Hardware interrupt: Hardware gate closing	In the activated state a hardware interrupt is generated when the hardware gate closes while the software gate is open.	• activated • deactivated	deactivated
Hardware interrupt: On reaching comparator	In the activated state a hardware interrupt is triggered on reaching the comparator (reaction) value. The process interrupt may only be released if in addition the value of "Characteristics of the output" is not "no comparison".	• activated • deactivated	deactivated
Hardware interrupt: Overflow	In the activated state a hardware interrupt is generated in the event of an overflow (exceeding the upper count limit).	• activated • deactivated	deactivated
Hardware interrupt: Underflow	In the activated state a hardware interrupt is generated in the event of an underflow (undershooting the lower count limit).	• activated • deactivated	deactivated

5.6 Counter - Controlling

Overview

The controlling of the appropriate counter is made from the user program by the SFB COUNT (SFB 47). The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.

With the SFB COUNT (SFB 47) you have following functional options:

- Start/Stop the counter via software gate SW_GATE
- Enable/control output DO
- Read the status bit
- Read the actual count and latch value
- Request to read/write internal counter registers

Parameter SFB 47

Name	Declaration	Data type	Address (Inst.-DB)	Default value	Comment
LADDR	INPUT	WORD	0.0	300h	This parameter is not evaluated. Always the internal I/O periphery is addressed.
CHANNEL	INPUT	INT	2.0	0	Channel number
SW_GATE	INPUT	BOOL	4.0	FALSE	Enables the Software gate
CTRL_DO	INPUT	BOOL	4.1	FALSE	Enables the output False: Standard Digital Output
SET_DO	INPUT	BOOL	4.2	FALSE	Parameter is not evaluated
JOB_REQ	INPUT	BOOL	4.3	FALSE	Initiates the job (edge 0-1)
JOB_ID	INPUT	WORD	6.0	0	Job ID
JOB_VAL	INPUT	DINT	8.0	0	Value for write jobs
STS_GATE	OUTPUT	BOOL	12.0	FALSE	Status of the internal gate
STS_STRT	OUTPUT	BOOL	12.1	FALSE	Status of the hardware gate (is only refreshed if "HW gate" is activated in hardware configuration before)
STS_LTCH	OUTPUT	BOOL	12.2	FALSE	Status of the latch input
STS_DO	OUTPUT	BOOL	12.3	FALSE	Status of the output
STS_C_DN	OUTPUT	BOOL	12.4	FALSE	Status of the down-count Always indicates the last direction of count. After the first SFB call STS_C_DN is set FALSE.
STS_C_UP	OUTPUT	BOOL	12.5	FALSE	Status of the up-count Always indicates the last direction of count. After the first SFB call STS_C_UP is set TRUE.
COUNTVAL		DINT	14.0	0	Actual count value
LATCHVAL		DINT	18.0	0	Actual latch value
JOB_DONE		BOOL	22.0	TRUE	New job can be started.
JOB_ERR		BOOL	22.1	FALSE	Job error
JOB_STAT		WORD	24.0	0	Job error ID

Local data only in instance DB

Name	Data type	Address (Instance DB)	Default value	Comment
RES00	BOOL	26.0	FALSE	reserved
RES01	BOOL	26.1	FALSE	reserved
RES02	BOOL	26.2	FALSE	reserved
STS_CMP	BOOL	26.3	FALSE	Comparator Status ^{*)} Status bit STS_CMP indicates that the comparison condition of the comparator is or was reached. STS_CMP also indicates that the output was set. (STS_DO = TRUE). This parameter is only refreshed if in the hardware configuration a comparison value is set at "Characteristics of the output".
RES04	BOOL	26.4	FALSE	reserved
STS_OFLW	BOOL	26.5	FALSE	Overflow status - is only set at range overflow ^{*)}
STS_UFLW	BOOL	26.6	FALSE	Underflow status - is only set at range underflow ^{*)}
STS_ZP	BOOL	26.7	FALSE	Status of the zero mark ^{*)} The bit is only set when counting without main direction. Indicates the zero mark. This is also set when the counter is set to 0 or if it starts counting.
JOB_OVAL	DINT	28.0		Output value for read request.
RES10	BOOL	32.0	FALSE	reserved
RES11	BOOL	32.1	FALSE	reserved
RES_STS	BOOL	32.2	FALSE	Reset status bits: Resets the status bits: STS_CMP, STS_OFLW, STS_ZP. The SFB must be twice to reset the status bit.

^{*)} Reset with RES_STS



Note!

Per channel you may call the SFB in each case with the same instance DB, since the data necessary for the internal operation are stored here. Writing accesses to outputs of the instance DB is not permissible.

Counter request interface

To read/write counter registers the request interface of the SFB 47 may be used.

So that a new job may be executed, the previous job must have been finished with JOB_DONE = TRUE.

Proceeding

The deployment of the request interface takes place at the following sequence:

- Edit the following input parameters:

Name	Data type	Address (DB)	Default	Comment
JOB_REQ	BOOL	4.3	FALSE	Initiates the job (edges 0-1)
JOB_ID	WORD	6.0	0	Job ID: 00h Job without function 01h Writes the count value 02h Writes the load value 04h Writes the comparison value 08h Writes the hysteresis 10h Writes the pulse duration 20h Writes the end value 82h Reads the load value 84h Reads the comparison value 88h Reads the hysteresis 90h Reads the pulse duration A0h Reads the end value
JOB_VAL	DINT	8.0	0	Value for write jobs (see table at the following page)

- Call the SFB. The job is processed immediately. JOB_DONE only applies to SFB run with the result FALSE. JOB_ERR = TRUE if an error occurred. Details on the error cause are indicated at JOB_STAT.

Name	Data type	Address (DB)	Default	Comment
JOB_DONE	BOOL	22.0	TRUE	New job can be started
JOB_ERR	BOOL	22.1	FALSE	Job error
JOB_STAT	WORD	24.0	0000h	Job error ID 0000h No error 0121h Compare value too low 0122h Compare value too high 0131h Hysteresis too low 0132h Hysteresis too high 0141h Pulse duration too low 0142h Pulse duration too high 0151h Load value too low 0152h Load value too high 0161h Count value too low 0162h Count value too high 01FFh Invalid job ID

- A new job may be started with JOB_DONE = TRUE.

- A value to be read of a read job may be found in JOB_OVAL in the instance DB at address 28.

Permitted value range for JOB_VAL

Continuous count

Job	Valid range
Writing counter directly	-2147483647 ($-2^{31}+1$) to +2147483646 ($2^{31}-2$)
Writing the load value	-2147483647 ($-2^{31}+1$) to +2147483646 ($2^{31}-2$)
Writing comparison value	-2147483648 (-2^{31}) to +2147483647 ($2^{31}-1$)
Writing hysteresis	0 to 255
Writing pulse duration*	0 to 510ms

Single/periodic count, no main count direction:

Job	Valid range
Writing counter directly	-2147483647 ($-2^{31}+1$) to +2147483646 ($2^{31}-2$)
Writing the load value	-2147483647 ($-2^{31}+1$) to +2147483646 ($2^{31}-2$)
Writing comparison value	-2147483648 (-2^{31}) to +2147483647 ($2^{31}-1$)
Writing hysteresis	0 to 255
Writing pulse duration*	0 to 510ms

Single/periodic count, main count direction up:

Job	Valid range
End value	2 to +2147483646 ($2^{31}-1$)
Writing counter directly	-2147483648 (-2^{31}) to end value -2
Writing the load value	-2147483648 (-2^{31}) to end value -2
Writing comparison value	-2147483648 (-2^{31}) to end value -1
Writing hysteresis	0 to 255
Writing pulse duration*	0 to 510ms

Single/periodic count, main count direction down:

Job	Valid range
Writing counter directly	2 to +2147483647 ($2^{31}-1$)
Writing the load value	2 to +2147483647 ($2^{31}-1$)
Writing comparison value	1 to +2147483647 ($2^{31}-1$)
Writing hysteresis	0 to 255
Writing pulse duration*	0 to 510ms

*) Only even values are allowed. Odd values are automatically rounded.

Latch function

As soon as during a count process an edge 0-1 is recognized at the "Latch" input of a counter, the recent counter value is stored in the according latch register. You may access the latch register via LATCHVAL of the SFB 47. A just in LATCHVAL loaded value remains after a STOP-RUN transition.

5.7 Counter – Functions

Parameterization

- Start the Siemens SIMATIC Manager with your project and open the hardware configurator.
- Place a profile rail.
- Configure at slot 2 the corresponding CPU from Siemens CPU 31xC.
- Open the dialog window "Properties" by a double click to the *Count* sub module of the CPU.
- As soon as an operating mode to the corresponding channel is selected, a dialog window for this operating mode is created and displayed and filled with default parameters.
- Execute the wished parameterization.
- Store the project with **Station** > *Save and compile*.
- Transfer the project to the CPU.

Load value, End value

Via the parameterization you have the opportunity to define a main counting direction for every counter. If "none" or "endless" is chosen, the complete counting range is available:

Limit counter	Valid value range
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 ($2^{31}-1$)

Otherwise this range may be limited in both directions by a start value as *load value* and an *end value*.

Main counting direction

Main counting direction forward

Upper restriction of the count range. The counter counts 0 res. load value in positive direction until the parameterized end value –1 and then jumps back to the load value with the next following encoder pulse.

Please note a load value may exclusively be set by the request interface of the counter.

Main counting direction backwards

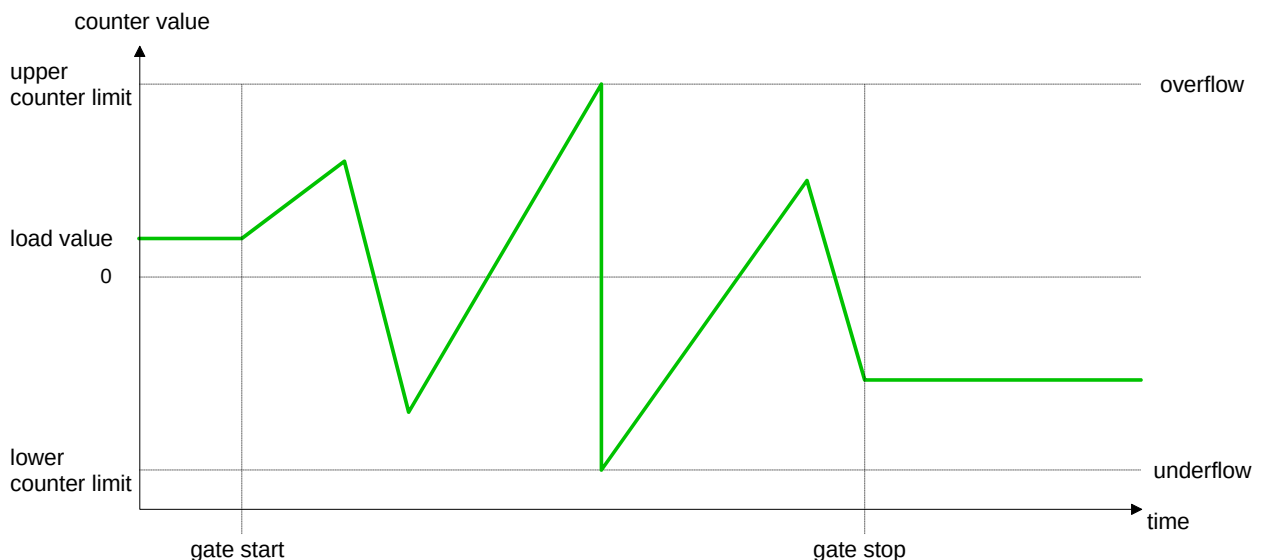
Lower restriction of the count range. The counter counts from the parameterized start- res. load value in negative direction to the parameterized end value +1 and then jumps back to the start value with the next following encoder pulse.

Please note an end value may exclusively be set by the request interface of the counter.

5.7.1 Count Continuously

In this operating mode, the counter counts from 0 res. from the load value. When the counter counts forward and reaches the upper count limit and another counting pulse in positive direction arrives, it jumps to the lower count limit and counts from there on. When the counter counts backwards and reaches the lower count limit and another counting pulse in negative direction arrives, it jumps to the upper count limit and counts from there on. The count limits are set to the maximum count range.

	Valid value range
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 ($2^{31}-1$)



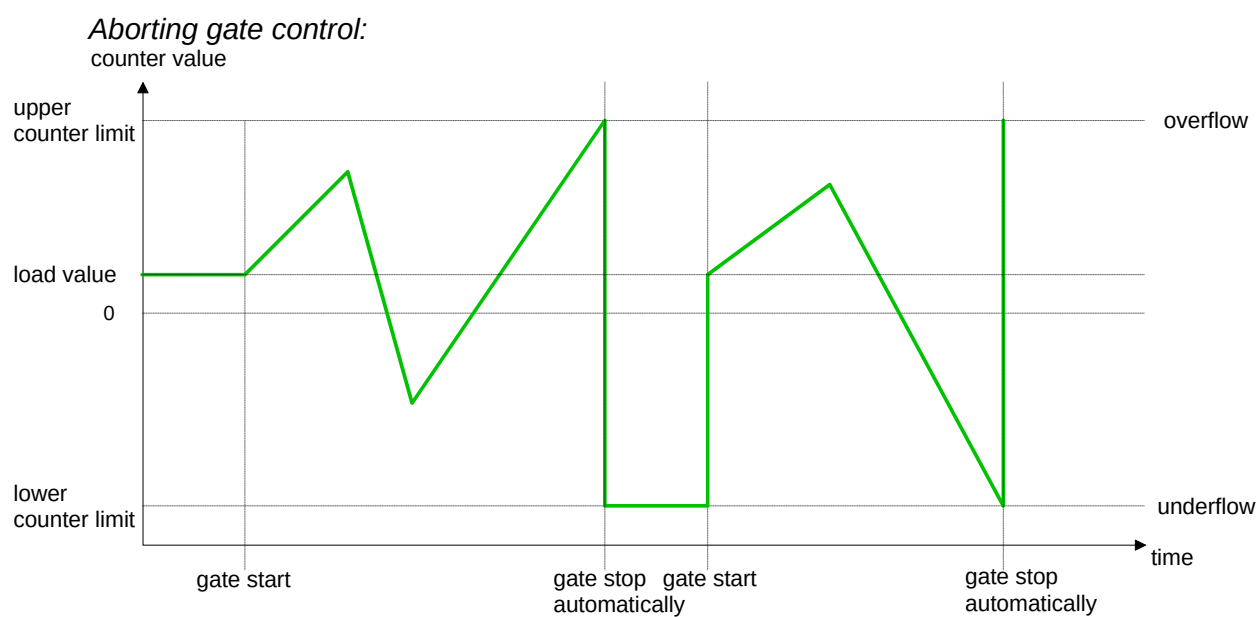
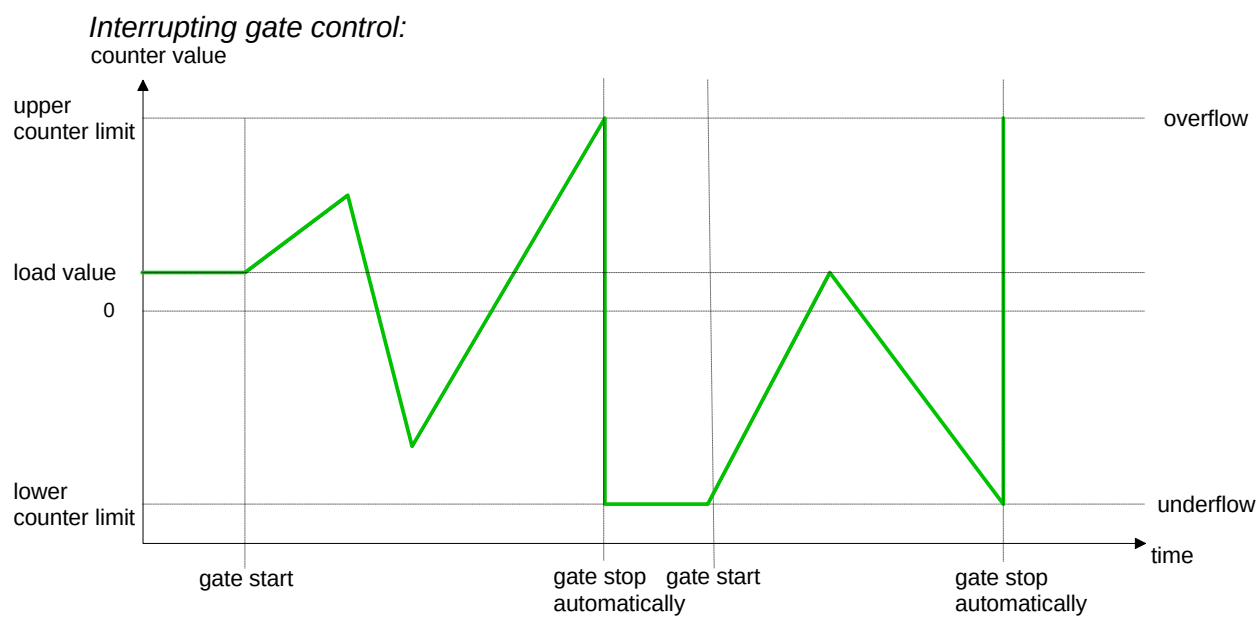
No main counting direction

- The counter counts once starting with the load value.

5.7.2 Count Once

- You may count forward or backwards.
- The count limits are set to the maximum count range.
- At over- or under run at the count limits, the counter jumps to the according other count limit and the gate is automatically closed.
- To restart the count process, you must create an edge 0-1 of the gate.
- At interrupting gate control, the count process continuous with the last recent counter value.
- At aborting gate control, the counter starts with the load value.

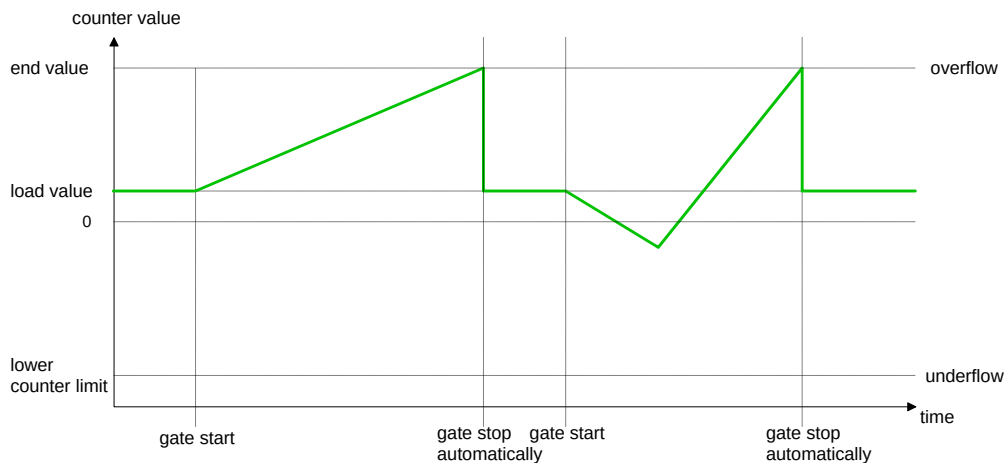
	Valid value range
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 ($2^{31}-1$)



Main counting direction forward

- The counter counts starting with the load value.
- When the counter reaches the end value -1 in positive direction, it jumps to the load value at the next positive count pulse and the gate is automatically closed.
- To restart the count process, you must create an edge 0-1 of the gate. The counter starts with the load value.

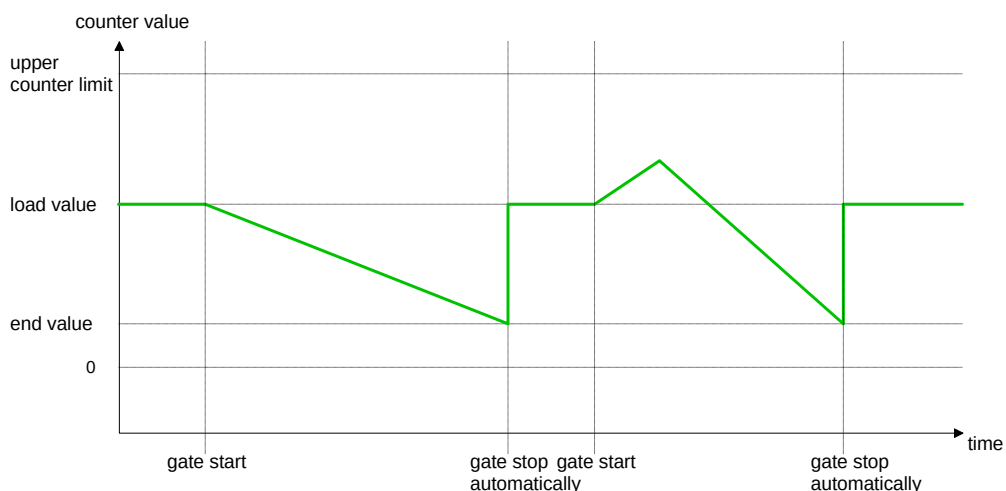
	Valid value range
Limit value	$-2\,147\,483\,646 (-2^{31}+1)$ to $+2\,147\,483\,646 (2^{31}-1)$
Lower count limit	-2^{31}



Main counting direction backwards

- The counter counts backwards starting with the load value.
- When the counter reaches the end value $+1$ in negative direction, it jumps to the load value at the next negative count pulse and the gate is automatically closed.
- To restart the count process, you must create an edge 0-1 of the gate. The counter starts with the load value.

	Valid value range
Limit value	$-2\,147\,483\,646 (-2^{31}+1)$ to $+2\,147\,483\,646 (2^{31}-1)$
Upper count limit	$+2^{31}-1$

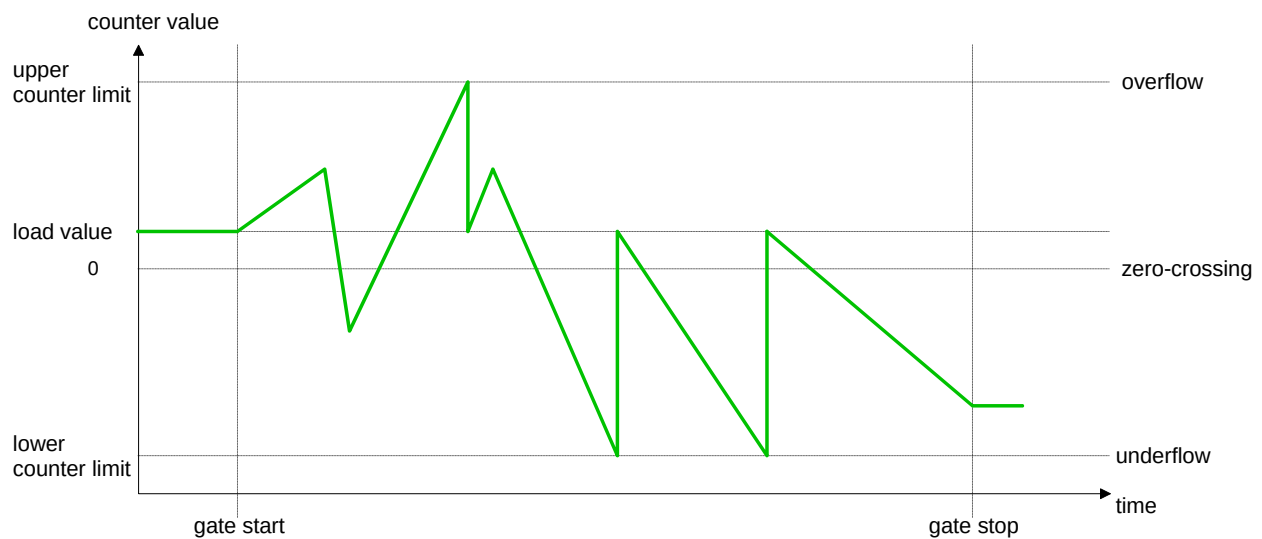


5.7.3 Count Periodically

No main counting direction

- The counter counts forward or backwards starting with the load value.
- At over- or under run at the count limits, the counter jumps to the according other count limit and counts from there on.
- The count limits are set to the maximum count range.

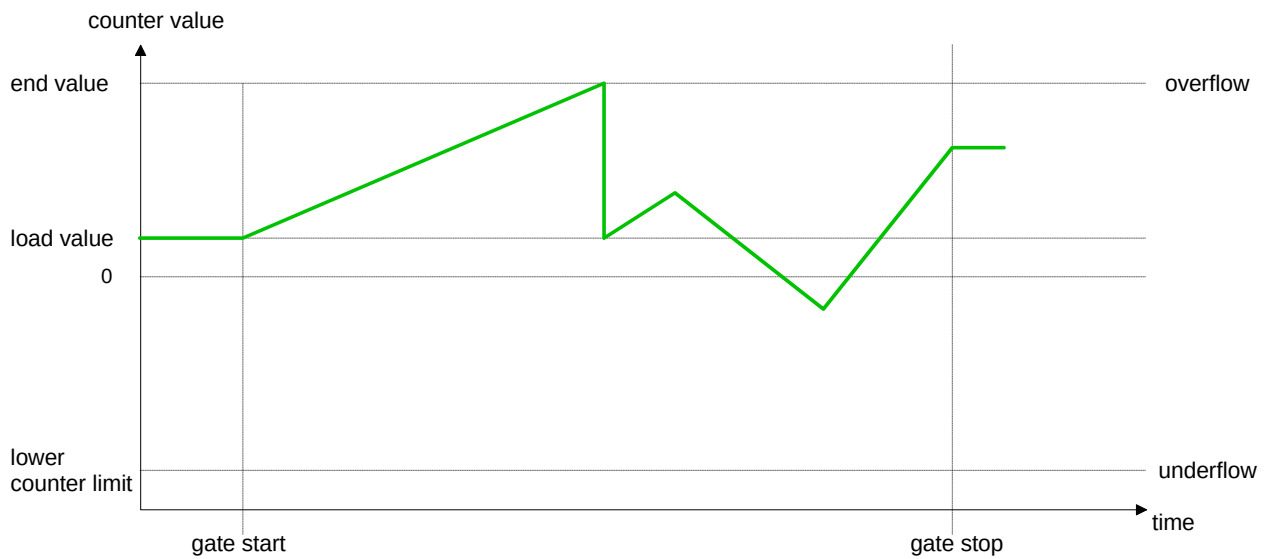
	Valid value range
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 ($2^{31}-1$)



Main counting direction forward

- The counter counts forward starting with the load value.
- When the counter reaches the end value -1 in positive direction, it jumps to the load value at the next positive count pulse.

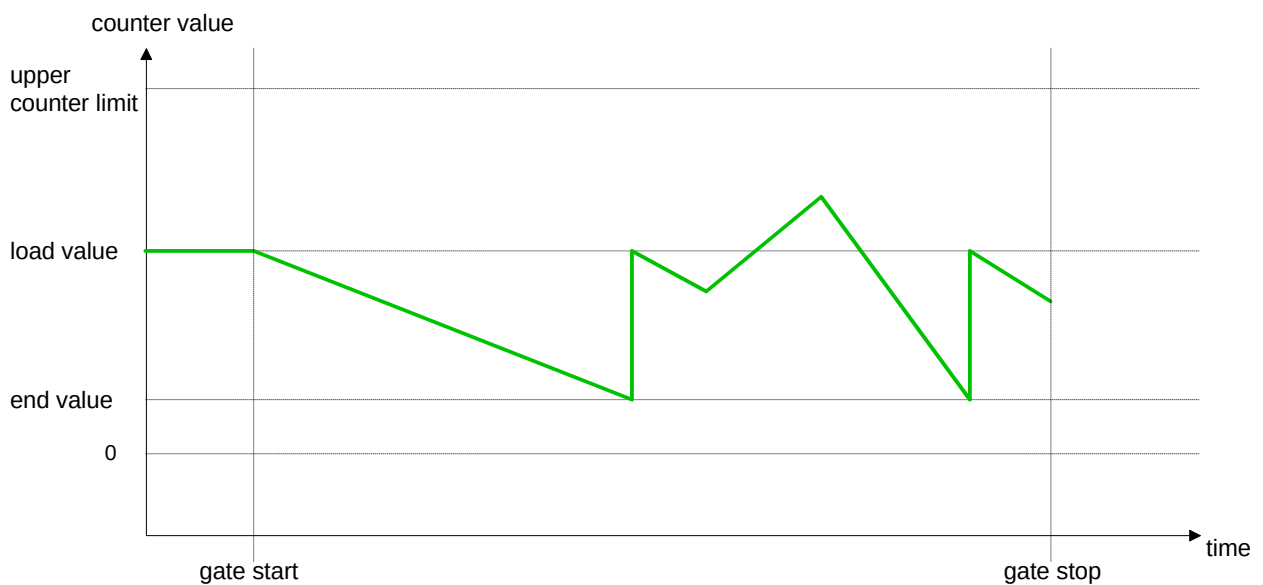
	Valid value range
Limit value	-2 147 483 647 ($-2^{31}+1$) to +2 147 483 647 ($2^{31}-1$)
Lower count limit	-2 147 483 648 (-2^{31})



Main counting direction backwards

- The counter counts backwards starting with the load value.
- When the counter reaches the end value+1 in negative direction, it jumps to the load value at the next negative count pulse.
- You may exceed the upper count limit.

	Valid value range
Limit value	$-2\,147\,483\,647 \cdot (-2^{31}+1)$ to $+2\,147\,483\,647 \cdot (2^{31}-2)$
Upper count limit	$+2\,147\,483\,647 \cdot (2^{31}-1)$



5.8 Counter - Additional functions

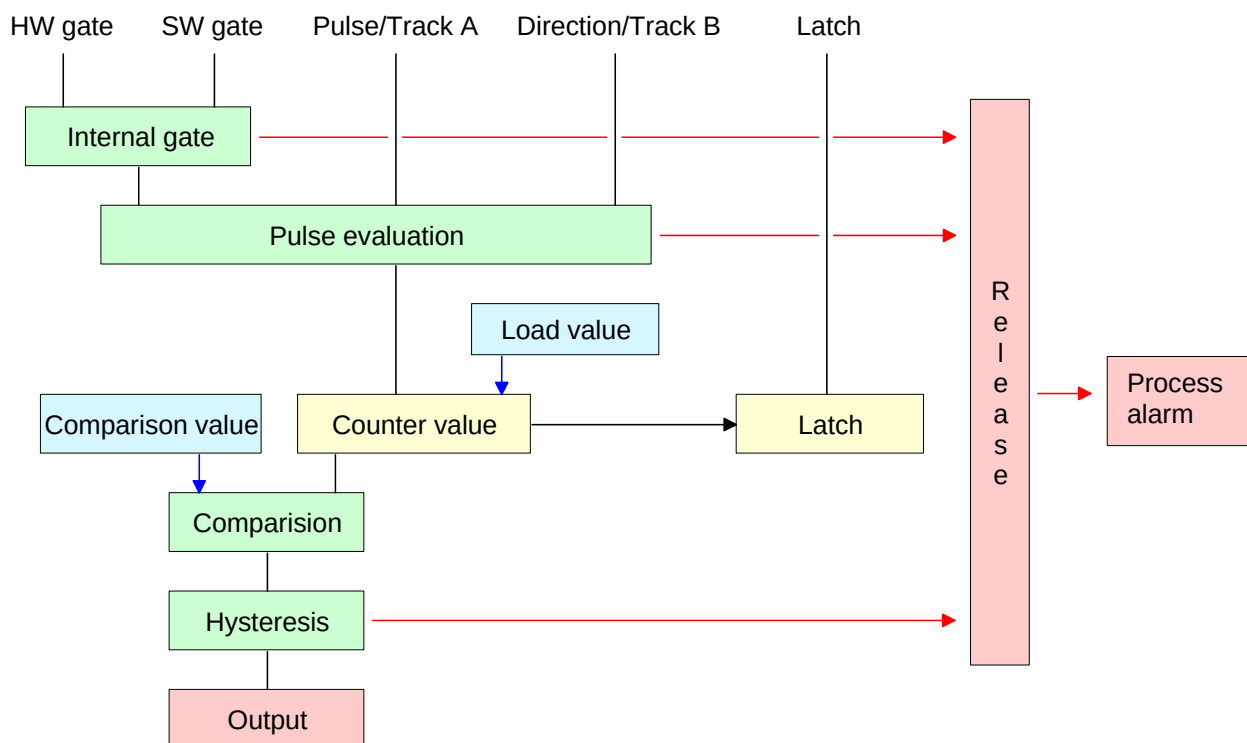
Overview

The following additional functions may be set via the parameterization for every counter:

- Gate function
The gate function serves the start, stop and interrupt of a count function.
- Latch function
An edge 0-1 at the digital input "Latch" stores the recent counter value in the latch register.
- Comparison
You may set a comparison value that activates res. de-activates a digital output res. releases a hardware interrupt depending on the counter value.
- Hysteresis
The setting of a hysteresis avoids for example a high output toggling when the value of an encoder signal shifts around a comparison value.

Schematic structure

The illustration shows how the additional functions influence the counting behavior. The following pages describe these functions in detail:



Gate function

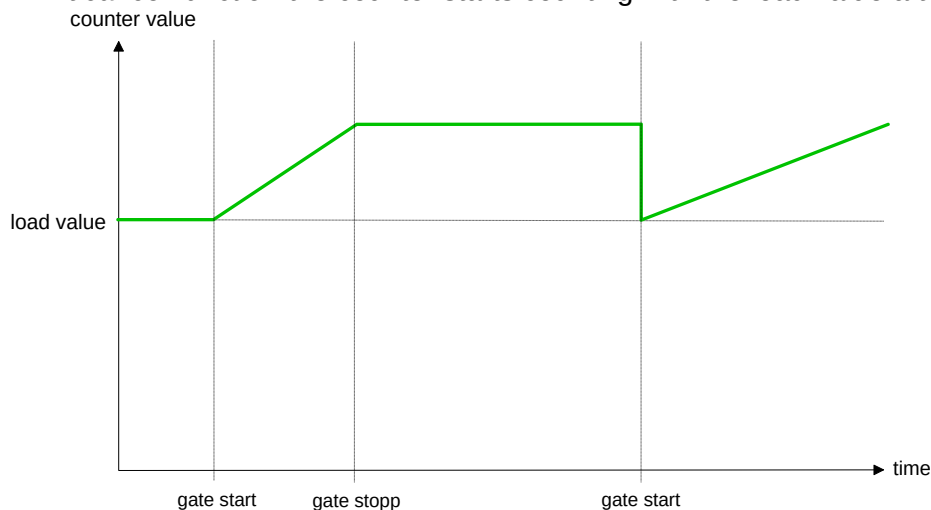
The counter is controlled by the internal gate (i gate). The i gate is the result of logic operation of hardware gate (HW gate) and software gate (SW gate), where the HW gate evaluation may be deactivated by the parameterization.

HW gate:	open (activate):	Edge 0-1 at hardware gate _x input of the module
	close (deactivate):	Edge 1-0 at hardware gate _x input of the module
SW gate:	open (activate):	In application program by setting SW_GATE of the SFB 47
	close (deactivate):	In application program by resetting SW_GATE of the SFB 47

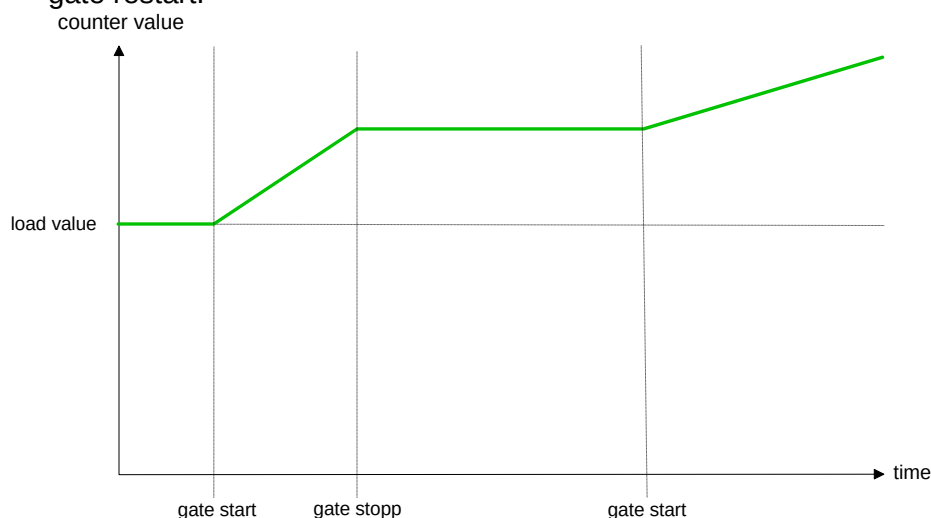
Gate functions cancel and stop

The parameterization defines if the gate cancels or stops the counter process.

- At *cancel function* the counter starts counting with the load value after gate restart.



- At *stop function*, the counter continues counting with the last recent counter value after gate restart.



Gate control abort, interruption

How the CPU should react at opening of the SW gate may be set with the parameter *Gate function*. The usage of the HW gate may be determined by the parameter *Hardware gate*.

Gate control via SW gate, canceling

(HW gate deactivated, gate function: Cancel count)

SW gate	HW gate	Reaction Counter
edge 0-1	de-activated	Restart with load value

Gate control via SW gate, stopping

(HW gate deactivated, gate function: Stop count)

SW gate	HW gate	Reaction Counter
edge 0-1	de-activated	Continue

Gate control via SW/HW gate, canceling

(HW gate activated, gate function: Cancel count)

SW gate	HW gate	Reaction Counter
edge 0-1	1	Continue
1	edge 0-1	Restart with load value

Gate control via SW/HW gate, stopping

(HW gate activated, gate function: Stop count)

SW gate	HW gate	Reaction Counter
edge 0-1	1	Continue
1	edge 0-1	Continue

Gate control "Count once"

Gate control via SW/HW gate, operating mode "Count once"

If the internal gate has been closed automatically it may only be opened again under the following conditions:

SW gate	HW gate	Reaction I gate
1	edge 0-1	1
edge 0-1 (after edge 0-1 at HW gate)	edge 0-1	1

Latch function

As soon as during a count process an edge 0-1 is recognized at the "Latch" input of a counter, the recent counter value is stored in the according latch register.

The latch value may be accessed by the parameter LATCHVAL of the SFB 47.

A just in LATCHVAL loaded value remains after a STOP-RUN transition.

Comparator

In the CPU a comparison value may be stored that is assigned to the digital output, to the status bit "Status Comparator" STS_CMP and to the hardware interrupt. The digital output may be activated depending on the count value and comparison value. A comparison value may be entered by the parameter assignment screen form respectively by the request interface of the SFB 47.

Characteristics of the output

You pre-define the behavior of the counter output via the parameterization:

- output never switches
- output switch when counter value $\geq$ comparison value
- output switch when counter value $\leq$ comparison value
- output switch at comparison value

No comparison

The output is set as normal output. The SFB input parameter CTRL_DO is effect less. The status bits STS_DO and STS_CMP (Status comparator in the instance DB) remain reset.

Count $\geq$ comparison value respectively Count $\leq$ comparison value

The output remains set as long as the counter value is higher or equal comparison value respectively lower or equal comparison value. For this the control bit must be set.

The comparison result is shown by the status bit STS_CMP.

This status bit may only be reset if the comparison condition is no longer fulfilled.

Pulse at comparison value

When the counter reaches the comparison value the output is set for the parameterized pulse duration. If you have configured a main count direction the output is only activated when the comparison value is reached with the specified main count direction. For this the control bit CTRL_DO should be set first.

The status of the digital output may be shown by the status bit ST_DO.

The comparison result is shown by the status bit STS_CMP. This status bit may only be reset if the pulse duration has run off. Comparison condition is no longer fulfilled.

With pulse time = 0 the output is as set as the comparison condition is fulfilled.

Pulse duration

For adaptation to the used actors a pulse duration may be specified. The pulse duration defines how long the output should be set. It may be preset in steps of 2ms between 0 and 510ms. The pulse duration starts with the setting of the according digital output. The inaccuracy of the pulse duration is less than 1ms.

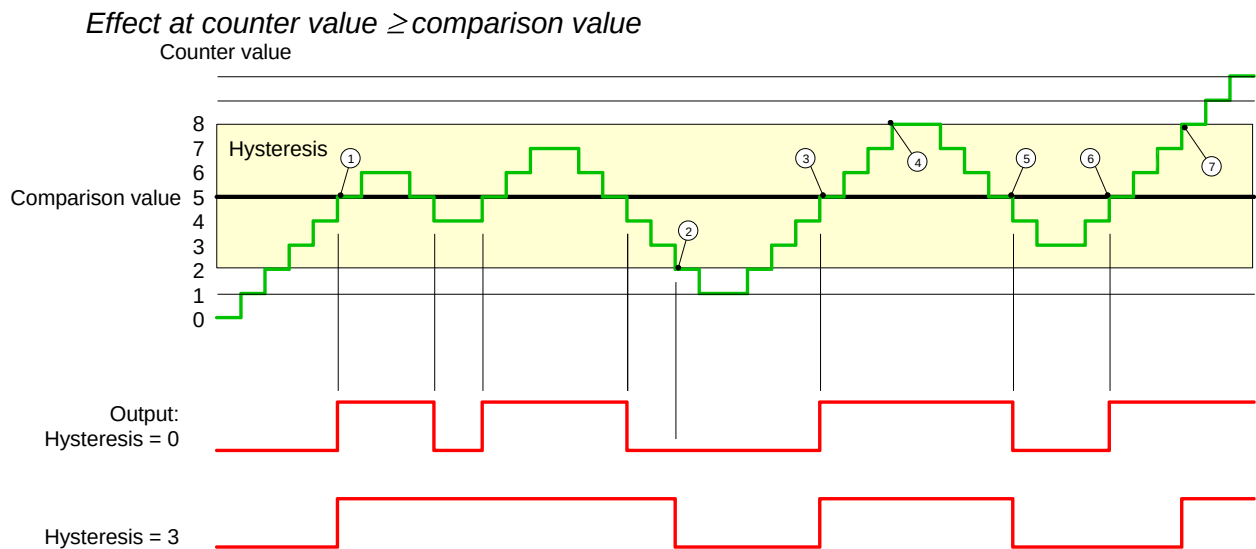
There is no past triggering of the pulse duration when the comparison value has been left and reached again during pulse output. A change of the pulse period during runtime is not applied until the next pulse.

Hysteresis

The hysteresis serves e.g. the avoidance of many toggle processes of the output and the interrupt, if the counter value is in the range of the comparison value. You may set a range of 0 to 255. The settings 0 and 1 deactivate the hysteresis. The hysteresis influences the zero run, over- and underflow.

An activated hysteresis remains active after a change. The new hysteresis range is taken over at the next reach of the comparison value.

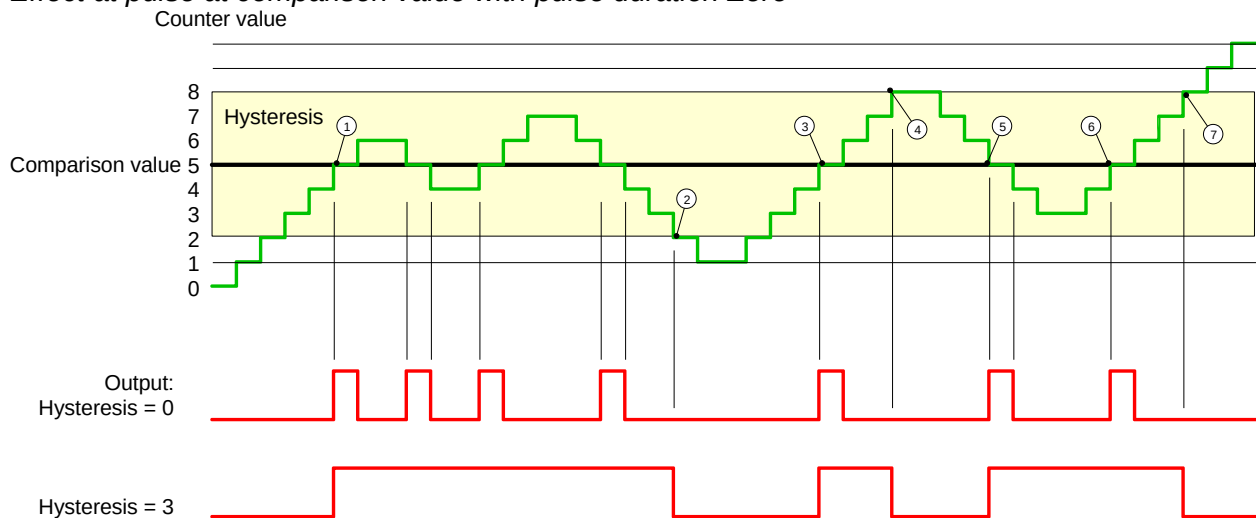
The following pictures illustrate the output behavior for hysteresis 0 and hysteresis 3 for the according conditions:



- ① Counter value $\geq$ comparison value $\rightarrow$ output is set and hysteresis activated
- ② Leave hysteresis range $\rightarrow$ output is reset
- ③ Counter value $\geq$ comparison value $\rightarrow$ output is set and hysteresis activated
- ④ Leave hysteresis range, output remains set for counter value $\geq$ comparison value
- ⑤ Counter value $<$ comparison value and hysteresis active $\rightarrow$ output is reset
- ⑥ Counter value $\geq$ comparison value $\rightarrow$ output is not set for hysteresis active
- ⑦ Leave hysteresis range, output remains set for counter value $\geq$ comparison value

With reaching the comparison condition the hysteresis gets active. At active hysteresis the comparison result remains unchanged until the counter value leaves the set hysteresis range. After leaving the hysteresis range a new hysteresis is only activated with again reaching the comparison conditions.

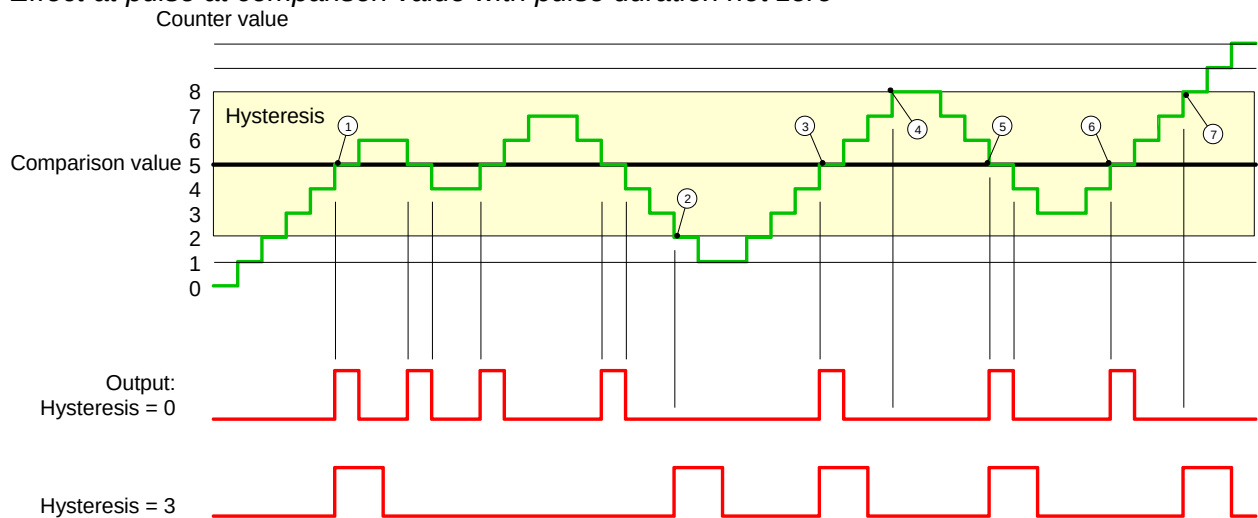
Effect at pulse at comparison value with pulse duration Zero



- ① Counter value = comparison value → output is set and hysteresis activated
- ② Leave hysteresis range → output is reset and counter value < comparison value
- ③ Counter value = comparison value → output is set and hysteresis activated
- ④ Output is reset for leaving hysteresis range and counter value > comparison value
- ⑤ Counter value = comparison value → output is set and hysteresis activated
- ⑥ Counter value = comparison value and hysteresis active → output remains set
- ⑦ Leave hysteresis range and counter value > comparison value → output is reset

With reaching the comparison condition the hysteresis gets active. At active hysteresis the comparison result remains unchanged until the counter value leaves the set hysteresis range. After leaving the hysteresis range a new hysteresis is only activated with again reaching the comparison conditions.

Effect at pulse at comparison value with pulse duration not zero



- ① Counter value = comparison value → pulse of the parameterized duration is put out, the hysteresis is activated and the counting direction stored
- ② Leaving the hysteresis range contrary to the stored counting direction → pulse of the parameterized duration is put out, the hysteresis is de-activated
- ③ Counter value = comparison value → pulse of the parameterized duration is put out, the hysteresis is activated and the counting direction stored
- ④ Leaving the hysteresis range without changing counting direction → hysteresis is de-activated
- ⑤ Counter value = comparison value → pulse of the parameterized duration is put out, the hysteresis is activated and the counting direction stored
- ⑥ Counter value = comparison value and hysteresis active → no pulse
- ⑦ Leaving the hysteresis range contrary to the stored counting direction → pulse of the parameterized duration is put out, the hysteresis is de-activated

With reaching the comparison condition the hysteresis gets active and a pulse of the parameterized duration is put out. As long as the counter value is within the hysteresis range, no other pulse is put out. With activating the hysteresis the counting direction is stored in the CPU. If the counter value leaves the hysteresis range contrary to the stored counting direction, a pulse of the parameterized duration is put out. Leaving the hysteresis range without direction change, no pulse is put out.

5.9 Counter - Diagnostic and interrupt

Overview

The parameterization allows you to define the following trigger for a hardware interrupt that may initialize a diagnostic interrupt:

- Status changes at an input (at opened SW gate)
- Status changes at the HW-gate (at opened SW gate)
- Reaching a comparison value
- Overflow respectively at overrun upper counter limit
- Underflow respectively at under run lower counter limit

Hardware interrupt

A hardware interrupt causes a call of the OB 40. Within the OB 40 you may find the logical basic address of the module that initialized the hardware interrupt by using the *Local word 6*. More detailed information about the initializing event is to find in the *local double word 8*.

Local double word 8 of the OB 40

The *local double word 8* of the OB 40 has the following structure:

Local byte	Bit 7 ... Bit 0
8	Bit 0: Edge at I+0.0 Bit 1: Edge at I+0.1 Bit 2: Edge at I+0.2 Bit 3: Edge at I+0.3 Bit 4: Edge at I+0.4 Bit 5: Edge at I+0.5 Bit 6: Edge at I+0.6 Bit 7: Edge at I+0.7
9	Bit 0: Edge at I+1.0 Bit 1: Edge at I+1.1 Bit 2: Edge at I+1.2 Bit 3: Edge at I+1.3 Bit 4: Edge at I+1.4 Bit 5: Edge at I+1.5 Bit 6: Edge at I+1.6 Bit 7: Edge at I+1.7
10	Bit 0: Gate counter 0 open (activated) Bit 1: Gate counter 0 closed Bit 2: Over-/underflow/end value counter 0 Bit 3: Counter 0 reached comparison value Bit 4: Gate counter 1 open (activated) Bit 5: Gate counter 1 closed Bit 6: Over-/underflow/ end value counter 1 Bit 7: Counter 1 reached comparison value
11	Bit 0: Gate counter 2 open (activated) Bit 1: Gate counter 2 closed Bit 2: Over-/underflow/end value counter 2

	Bit 3: Counter 2 reached comparison value Bit 7 ... 4: reserved
--	--

Diagnostic interrupt

Via the parameterization (record set 7Fh) you may activate a global diagnostic interrupt for the analog and digital part.

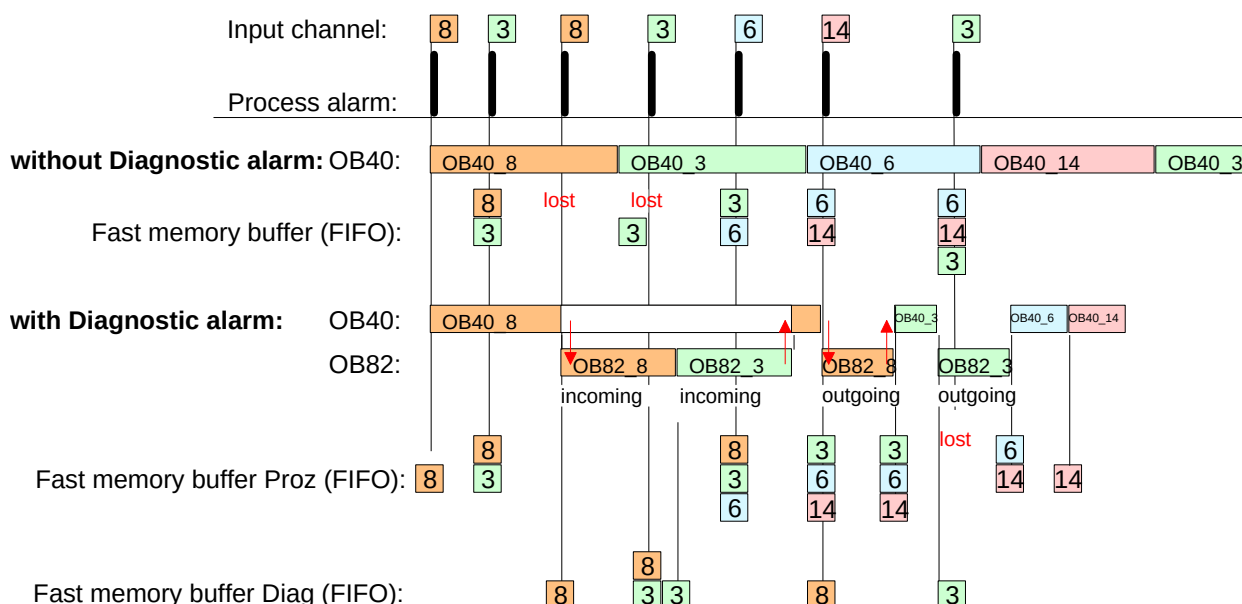
A diagnostic interrupt occurs when during a hardware interrupt execution in OB 40 another hardware interrupt is thrown for the same event. The initialization of a diagnostic interrupt interrupts the recent hardware interrupt execution in OB 40 and branches in OB 82 to diagnostic interrupt processing_{incoming}. If during the diagnostic interrupt processing other events are occurring at other channels that may also cause a process res. diagnostic interrupt, these are interim stored.

After the end of the diagnostic interrupt processing at first all interim-stored diagnostic interrupts are processed in the sequence of their occurrence and then all hardware interrupts.

If a channel where currently a diagnostic interrupt_{incoming} is processed res. interim stored initializes further hardware interrupts, these get lost. When a hardware interrupt for which a diagnostic interrupt_{incoming} has been released is ready, the diagnostic interrupt processing is called again as diagnostic interrupt_{going}.

All events of a channel between diagnostic interrupt_{incoming} and diagnostic interrupt_{going} are not stored and get lost. Within this time window (1st diagnostic interrupt_{incoming} until last diagnostic interrupt_{going}) the SF-LED of the CPU is on. Additionally for every diagnostic interrupt_{incoming/going} an entry in the diagnostic buffer of the CPU occurs.

Example



Diagnostic interrupt processing

Every OB 82 call causes an entry in the diagnostic buffer of the CPU containing error cause and module address.

By using the SFC 59 you may read the diagnostic bytes.

At de-activated diagnostic interrupt you have access to the last recent diagnostic event.

If you've activated the diagnostic function in your hardware configuration, the contents of record set 0 are already in the local double word 8 when calling the OB 82. The SFC 59 allows you to also read the record set 1 that contains additional information.

After leaving the OB 82 a clear assignment of the data to the last diagnostic interrupt is not longer possible.

Record set 0 Diagnostic_{incoming}

The record sets of the diagnostic range have the following structure:

Byte	Bit 7 ... 0
0	Bit 0: set at module failure Bit 1: 0 (fix) Bit 2: set at external error Bit 3: set at channel error Bit 4: set when external auxiliary supply is missing Bit 7 ... 5: 0 (fix)
1	Bit 3 ... 0: Module class 0101b: Analog 1111b: Digital Bit 4: Channel information present Bit 7 ... 5: 0 (fix)
2	Bit 3 ... 0: 0 (fix) Bit 4: Failure module internal supply voltage (output overload) Bit 7 ... 5: 0 (fix)
3	Bit 5 ... 0: 0 (fix) Bit 6: Hardware interrupt lost Bit 7: 0 (fix)

Record set 0 Diagnostic_{going}

After the removing error a diagnostic message_{going} takes place if the diagnostic interrupt release is still active.

Record set 0 (Byte 0 to 3):

Byte	Bit 7 ... 0
0	Bit 0: set at module failure Bit 1: 0 (fix) Bit 2: set at external error Bit 3: set at channel error Bit 4: set when external auxiliary supply is missing Bit 7 ... 5: 0 (fix)
1	Bit 3 ... 0: Module class 0101b: Analog module 1111b: Digital Bit 4: Channel information present Bit 7 ... 5: 0 (fix)
2	00h (fix)
3	00h (fix)

Diagnostic Record set 1 (Byte 0 ... 15)

The record set 1 contains the 4Byte of the record set 0 and additionally 12Byte module specific diagnostic data.

The diagnostic bytes have the following assignment:

Byte	Bit 7 ... 0
0 ... 3	Contents record set 0 (see page before)
4	Bit 6 ... 0: channel type (here 70h) 70h: Digital input 71h: Analog input 72h: Digital output 73h: Analog output 74h: Analog in-/output Bit 7: More channel types present 0: no 1: yes
5	Number of diagnostic bits per channel (here 08h)
6	Number of channels of a module (here 08h)
7	Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) Bit 3: Error in channel group 3 (I+1.4 ... I+1.7) Bit 4: Error in channel group 4 (Counter 0) Bit 5: Error in channel group 5 (Counter 1) Bit 6: Error in channel group 6 (Counter 2) Bit 7: reserved
8	Diagnostic interrupt due to hardware interrupt lost at... Bit 0: ... input I+0.0 Bit 1: 0 (fix) Bit 2: ... input I+0.1 Bit 3: 0 (fix) Bit 4: ... input I+0.2 Bit 5: 0 (fix) Bit 6: ... input I+0.3 Bit 7: 0 (fix)
9	Diagnostic interrupt due to hardware interrupt lost at... Bit 0: ... input I+0.4 Bit 1: 0 (fix) Bit 2: ... input I+0.5 Bit 3: 0 (fix) Bit 4: ... input I+0.6 Bit 5: 0 (fix) Bit 6: ... input I+0.7 Bit 7: 0 (fix)

Dual Port Memory Interface

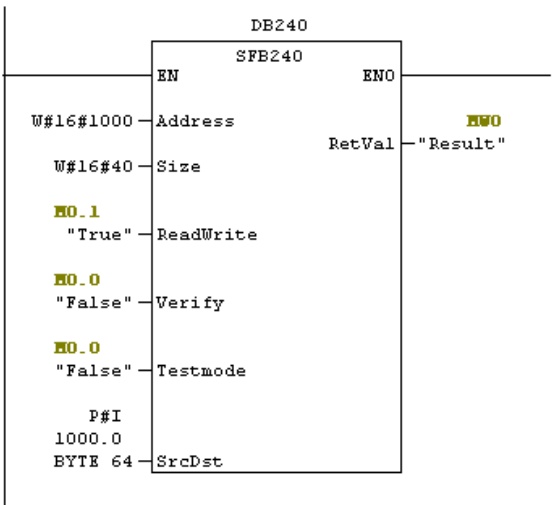
Byte	Bit 7 ... 0
10	Diagnostic interrupt due to hardware interrupt lost at... Bit 0: ... input I+1.0 Bit 1: 0 (fix) Bit 2: ... input I+1.1 Bit 3: 0 (fix) Bit 4: ... input I+1.2 Bit 5: 0 (fix) Bit 6: ... input I+1.3 Bit 7: 0 (fix)
11	Diagnostic interrupt due to hardware interrupt lost at... Bit 0: ... input I+1.4 Bit 1: 0 (fix) Bit 2: ... input I+1.5 Bit 3: 0 (fix) Bit 4: ... input I+1.6 Bit 5: 0 (fix) Bit 6: ... input I+1.7 Bit 7: 0 (fix)
12	Diagnostic interrupt due to hardware interrupt lost at... Bit 0: ... Gate Counter 0 closed Bit 1: 0 (fix) Bit 2: ... Gate Counter 0 open Bit 3: 0 (fix) Bit 4: ... Over-/underflow/end value Counter 0 Bit 5: 0 (fix) Bit 6: ... Counter 0 reached comparison value Bit 7: 0 (fix)
13	Diagnostic interrupt due to hardware interrupt lost at... Bit 0: ... Gate Counter 1 closed Bit 1: 0 (fix) Bit 2: ... Gate Counter 1 open Bit 3: 0 (fix) Bit 4: ... Over-/underflow/end value Counter 1 Bit 5: 0 (fix) Bit 6: ... Counter 1 reached comparison value Bit 7: 0 (fix)
14	Diagnostic interrupt due to hardware interrupt lost at... Bit 0: ... Gate Counter 2 closed Bit 1: 0 (fix) Bit 2: ... Gate Counter 2 open Bit 3: 0 (fix) Bit 4: ... Over-/underflow/end value Counter 2 Bit 5: 0 (fix) Bit 6: ... Counter 2 reached comparison value Bit 7: 0 (fix)

6 Dual Port Memory Interface

User Interface:

The SFB 240 (System Function Block) provides access to the DPR (Dual-Port-RAM) Interface. There are 4096 Byte input and 4096 Byte output data.

Example:



Netzwerk 2 : Titel:

Kommentar:



Netzwerk 3 : Titel:

Kommentar:



Dual Port Memory Interface

Parameters:

WORD: 16 Bit Integer
BOOL: 1 Bit
S7-Any-Pointer: Please see S7 documentation for details.

Type:	Name [Range]	Description
IN WORD	Address: [0..4095]	Address within area (seperate for input and output)
IN BOOL	ReadWrite :[0..1]	Selects read(0) or write(1) (different areas in DPR)
INOUT S7-Any-Pointer	SrcDst: [I,Q,M]	Source or destination area in S7-CPU (S7-Pointer)
IN WORD	Size: [1..4095]	Data size (same as in Any-Pointer)
IN BOOL	Verify: [0..1]	For test: verify written data
IN BOOL	Testmode: [0..1]	For test: (1) write into the input area (loopback)
OUT WORD	RetVal	Return Value

Technical Background:

Input area: Access to the DPR address 0x1000++0xffff (4096 Bytes)
Output area: Access to the DPR Address 0x2000++0xffff (4096 Bytes)

There is no option for IRQs!

Special Return Values (for other Return Values see CPU-Manual):

- 0 No Error (Successful executed)
- 8001 Address not valid [0..4096]
- 8002 Wrong SrcDst Any-Pointer (only I,Q,M Area!)
- 8003 Size in Any-Pointer and Size Parameter do not match
- 8004 Size exceeds range >4096
- 8005 Address+size >4096
- 8006 SrcDst: there is a BitAddress $\neq 0$ within the AnyPointer
- 8007 Only if Verify-Parameter = „1“: There was an Error during data verification

Comments on Address and Size

DWORD (32 Bit Copy): When Address modulo 4 = 0 and Pointer modulo 4 = 0
WORD (16 Bit Copy): When Address modulo 2 = 0 and Pointer modulo 2 = 0

Any other Case: Byte Copy (not recommended)

We recommend using the RUN/STOP Signal to disable all Outputs in case of CPU-Stop.

The RUN/Stop Signal will go into RUN-State AFTER the OB100 (Startup-Block).
The Dual Port Memory should be initialized within this OB100 to avoid unexpected behavior.

7 PtP Communication

Overview

Content of this chapter is the deployment of the RS485 slot for serial PtP communication. Here you'll find all information about the protocols and project engineering of the interface, which are necessary for the serial communication using the RS485 interface.

7.1 Brief introduction

General

The RS485 interface X3 of the CPU 313-6CF03 may be set to PtP communication (**point-to-point**) by means of a hardware configuration.

The RS485 interface in PtP operation supports the serial process connection to different source or destination systems.

Protocols

The protocols res. procedures ASCII, STX/ETX, 3964R, USS and Modbus are supported.

Parameterization

The parameterization happens during runtime using the SFC 216 (SER_CFG). For this you have to store the parameters in a DB for all protocols except ASCII.

Communication

The communication is controlled by SFCs. Send takes place via SFC 217 (SER_SND) and receive via SFC 218 (SER_RCV).

The repeated call of the SFC 217 SER_SND delivers a return value for 3964R, USS and Modbus via RetVal that contains, among other things, recent information about the acknowledgement of the partner station.

USS and Modbus protocols allow evaluation of the receipt telegram by calling the SFC 218 SER_RCV after SER_SND.

The SFCs are included in the consignment of the CPU.

Overview SFCs for serial communication

The following SFCs are used for the serial communication:

SFC		Description
SFC 216	SER_CFG	RS485 parameterize
SFC 217	SER_SND	RS485 send
SFC 218	SER_RCV	RS485 receive

7.2 Protocols and procedures

Overview

The CPU supports the following protocols and procedures:

- ASCII communication
- STX/ETX
- 3964R
- USS
- Modbus

ASCII

ASCII data communication is one of the simple forms of data exchange.

Incoming characters are transferred 1 to 1.

At ASCII, with every cycle the read-SFC is used to store the data that is in the buffer at request time in a parameterized receive data block. If a telegram is spread over various cycles, the data is overwritten. There is no reception acknowledgement. The communication procedure has to be controlled by the concerning user application.

An according Receive_ASCII-FB is available.

STX/ETX

STX/ETX is a simple protocol with start and end ID, where STX stands for **S**tart of **T**ext and ETX for **E**nd of **T**ext.

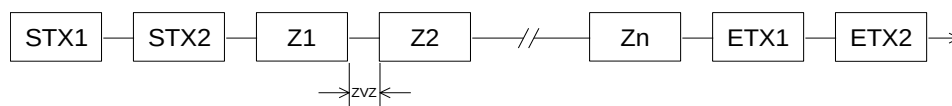
The STX/ETX procedure is suitable for the transfer of ASCII characters. It does not use block checks (BCC). Any data transferred from the periphery must be preceded by a start followed by the data characters and the end character.

Depending of the byte width the following ASCII characters can be transferred: 5Bit: not allowed; 6Bit: 20...3Fh, 7Bit: 20...7Fh, 8Bit: 20...FFh.

The effective data which includes all the characters between Start and End are transferred to the PLC when the End has been received.

When data is send from the PLC to a peripheral device, any user data is handed to the SFC 217 (SER_SND) and is transferred with added Start- and End-ID to the communication partner.

Message structure:



You may define up to 2 Start- and End-IDs.

You may work with 1, 2 or no Start- and with 1, 2 or no End-ID. As Start- res. End-ID all Hex values from 01h to 1Fh are permissible. Characters above 1Fh are ignored. In the user data, characters below 20h are not allowed and may cause errors. The number of Start- and End-IDs may be different (1 Start, 2 End res. 2 Start, 1 End or other combinations). If no End-ID is

defined, all read characters are transferred to the PLC after a parameterized character delay time (Timeout).

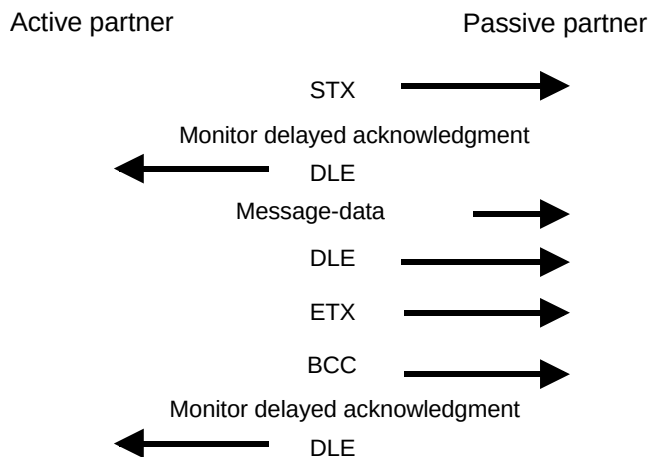
3964R

The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free receipt.

The procedure employs the following control characters:

- STX **S**tart of **T**ext
- DLE **D**ata **L**ink **E**scape
- ETX **E**nd of **T**ext
- BCC **B**lock **C**heck **C**haracter
- NAK **N**egative **A**cknowledge

Procedure



You may transfer a maximum of 255Byte per message.



Note!

When a DLE is transferred as part of the information it is repeated to distinguish between data characters and DLE control characters that are used to establish and to terminate the connection (DLE duplication). The DLE duplication is reversed in the receiving station.

The 3964R procedure requires that a lower priority is assigned to the communication partner. When communication partners issue simultaneous send commands, the station with the lower priority will delay its send command.

USS

The USS protocol (**U**niverselle **s**erielle **S**chnittstelle = universal serial interface) is a serial transfer protocol defined by Siemens for the drive and system components. This allows to build-up a serial bus connection between a superordinated master and several slave systems.

The USS protocol enables time cyclic telegram traffic by presetting a fix telegram length.

The following features characterize the USS protocol:

- Multi point connection
- Master-Slave access procedure
- Single-Master-System
- Max. 32 participants
- Simple and secure telegram frame

You may connect 1 master and max. 31 slaves at the bus where the single slaves are addressed by the master via an address sign in the telegram. The communication happens exclusively in half-duplex operation.

After a send command, the acknowledgement telegram must be read by a call of the SFC 218 SER_RCV.

The telegrams for send and receive have the following structure:

Master-Slave telegram

STX	LGE	ADR	PKE		IND		PWE		STW		HSW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

Slave-Master telegram

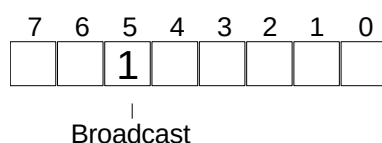
STX	LGE	ADR	PKE		IND		PWE		ZSW		HIW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

Where

STX: Start sign
LGE: Telegram length
ADR: Address
PKE: Parameter ID
IND: Index
PWE: Parameter value

STW: Control word
ZSW: State word
HSW: Main set value
HIW: Main effective value
BCC: Block Check Character

Broadcast by setting Bit 5 in ADR-Byte



A request may be directed to a certain slave or be sent to all slaves as broadcast message. For the identification of a broadcast message you have to set Bit 5 to 1 in the ADR-Byte. Here the slave addr. (Bit 0 ... 4) is ignored. In opposite to a "normal" send command, the broadcast does not require a telegram evaluation via SFC 218 SER_RCV. Only write commands may be sent as broadcast.

Modbus

The Modbus protocol is a communication protocol that fixes a hierarchic structure with one master and several slaves.

Physically, Modbus works with a serial half-duplex connection.

There are no bus conflicts occurring, because the master can only communicate with one slave at a time. After a request from the master, this waits for a preset delay time for an answer of the slave. During the delay time, communication with other slaves is not possible.

After a send command, the acknowledgement telegram must be read by a call of the SFC 218 SER_RCV.

The request telegrams send by the master and the respond telegrams of a slave have the following structure:

Start sign	Slave address	Function Code	Data	Flow control	End sign
------------	---------------	---------------	------	--------------	----------

Broadcast with slave address = 0

A request can be directed to a special slave or at all slaves as broadcast message. To mark a broadcast message, the slave address 0 is used.

In opposite to a "normal" send command, the broadcast does not require a telegram evaluation via SFC 218 SER_RCV.

Only write commands may be sent as broadcast.

ASCII, RTU mode

Modbus offers 2 different transfer modes:

- ASCII mode: Every Byte is transferred in the 2 sign ASCII code. The data are marked with a start and an end sign. This causes a transparent but slow transfer.
- RTU mode: Every Byte is transferred as one character. This enables a higher data pass through as the ASCII mode. Instead of start and end sign, a time control is used.

The mode selection happens during runtime by using the SFC 216 SER_CFG.

Supported Modbus protocols

The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

7.3 Principals of the data transfer

Overview

The data transfer is handled during runtime by using SFCs. The principle of data transfer is the same for all protocols and is shortly illustrated in the following.

Principle

Data that is into the according data channel by the PLC, is stored in a FIFO send buffer (first in first out) with a size of 2x1024Byte and then put out via the interface.

When the interface receives data, this is stored in a FIFO receive buffer with a size of 2x1024Byte and can there be read by the PLC.

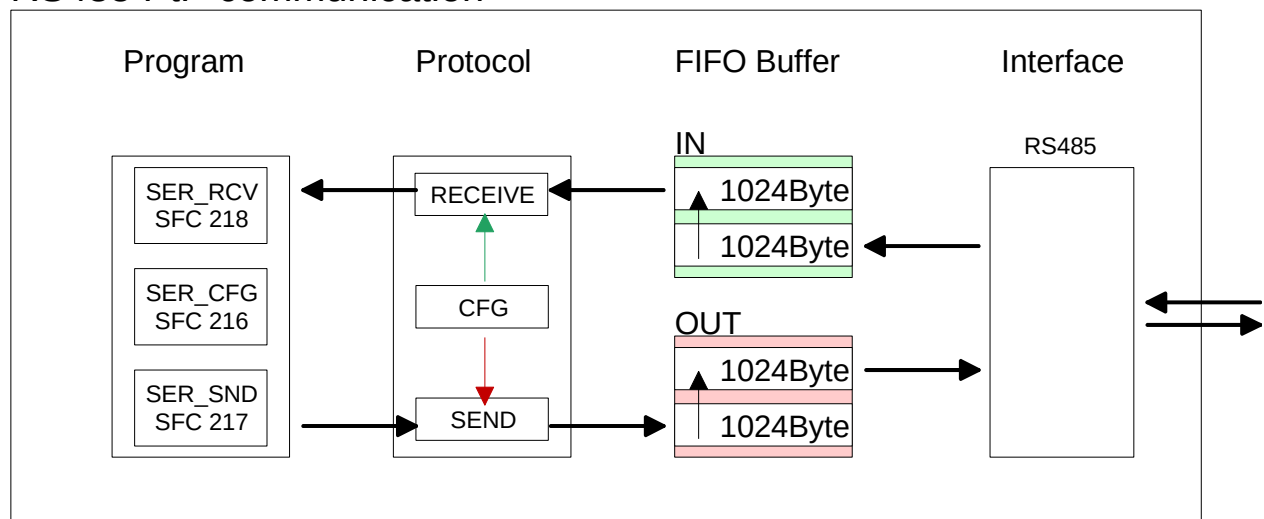
If the data is transferred via a protocol, the adoption of the data to the according protocol happens automatically.

In opposite to ASCII and STX/ETX, the protocols 3964R, USS and Modbus require the acknowledgement of the partner.

An additional call of the SFC 217 SER_SND causes a return value in RetVal that includes among others recent information about the acknowledgement of the partner.

Further on for USS and Modbus after a SER_SND the acknowledgement telegram must be evaluated by call of the SFC 218 SER_RCV.

RS485 PtP communication



7.4 RS485 interface

Properties RS485

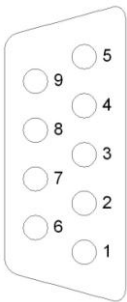
The RS485 interface X3 of the CPU 313-6CF03 may be set to PtP communication (point-to-point) by means of a hardware configuration.

Parameterization and communication happens by means of SFCs. The following characteristics distinguish the RS485 interface:

- Logical states represented by voltage differences between the two cores of a twisted pair cable
- Serial bus connection in two-wire technology using half duplex mode
- Data communications up to a max. distance of 500m
- Data communication rate up to 115.2kBaud

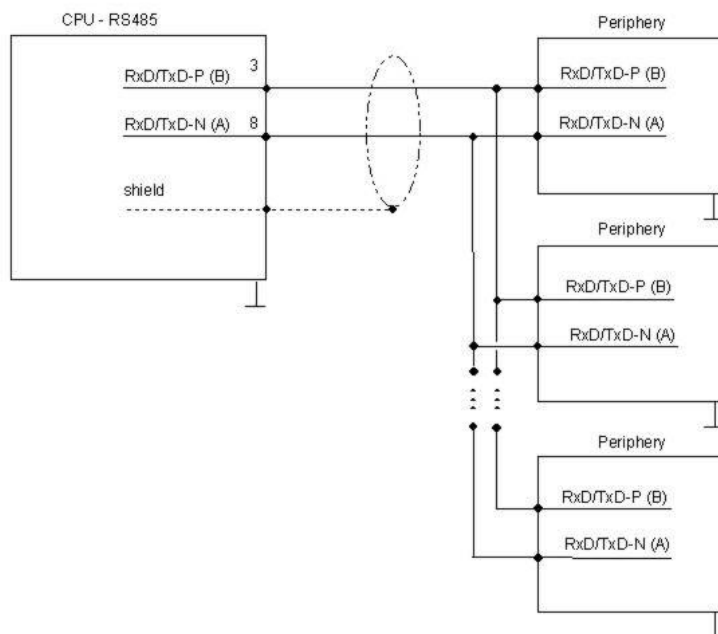
Connection RS485

9pin SubD jack



Pi n	RS485
1	n.c.
2	M24V
3	RxD/TxD-P (Line B)
4	RTS
5	M5V
6	P5V
7	P24V
8	RxD/TxD-N (Line A)
9	n.c.

Connection



Parameterization

SFC 216 (SER_CFG)

The parameterization happens during runtime deploying the SFC 216 (SER_CFG). You have to store the parameters for STX/ETX, 3964R, USS and Modbus in a DB.

Name	Declaration	Type	Comment
Protocol	IN	BYTE	1=ASCII, 2=STX/ETX, 3=3964R
Parameter	IN	ANY	Pointer to protocol-parameters
Baudrate	IN	BYTE	Number of the baudrate
CharLen	IN	BYTE	0=5Bit, 1=6Bit, 2=7Bit, 3=8Bit
Parity	IN	BYTE	0=None, 1=Odd, 2=Even
StopBits	IN	BYTE	1=1Bit, 2=1.5Bit, 3=2Bit
FlowControl	IN	BYTE	1 (fix)
RetVal	OUT	WORD	Return value (0 = OK)

Parameter description

All time settings for timeouts must be set as hexadecimal value. Find the hex value by multiply the wanted time in seconds with the baudrate.

Example: Wanted time 8ms at a baudrate of 19200Baud
 Calculation: $19200\text{Bit/s} \times 0,008\text{s} \approx 154\text{Bit} \rightarrow (9\text{Ah})$
 Hex value is 9Ah.

Protocol

Here you fix the protocol to be used. You may choose between:

- 1: ASCII
- 2: STX/ETX
- 3: 3964R
- 4: USS Master
- 5: Modbus RTU Master
- 6: Modbus ASCII Master

Parameter (as DB)

At ASCII protocol, this parameter is ignored.

At STX/ETX, 3964R, USS and Modbus you fix here a DB that contains the communication parameters and has the following structure for the according protocols:

Data block at STX/ETX

DBB0:	STX1	BYTE	(1. Start-ID in hexadecimal)
DBB1:	STX2	BYTE	(2. Start-ID in hexadecimal)
DBB2:	ETX1	BYTE	(1. End-ID in hexadecimal)
DBB3:	ETX2	BYTE	(2. End-ID in hexadecimal)
DBW4:	TIMEOUT	WORD	(max. delay time between 2 telegrams)

**Note!**

The start res. end sign should always be a value <20, otherwise the sign is ignored!

Data block at 3964R

DBB0:	Prio	BYTE	(The priority of both partners must be different)
DBB1:	ConnAttmptNr	BYTE	(Number of connection trials)
DBB2:	SendAttmptNr	BYTE	(Number of telegram retries)
DBW4:	CharTimeout	WORD	(Character delay time)
DBW6:	ConfTimeout	WORD	(Acknowledgement delay time)

Data block at USS

DBW0:	Timeout	WORD	(Delay time in)
-------	---------	------	-----------------

Data block at Modbus-Master

DBW0:	Timeout	WORD	(Respond delay time)
-------	---------	------	----------------------

Baud rate

Velocity of data transfer in Bit/s (Baud).

04h: 1200Baud	05h: 1800Baud	06h: 2400Baud	07h: 4800Baud
08h: 7200Baud	09h: 9600Baud	0Ah: 14400Baud	0Bh: 19200Baud
0Ch: 38400Baud	0Dh: 57600Baud	0Eh: 115200Baud	

CharLen

Number of data bits where a character is mapped to.

0: 5Bit 1: 6Bit 2: 7Bit 3: 8Bit

Parity

The parity is -depending on the value- even or odd. For parity control, the information bits are extended with the parity bit that amends via its value „0“ or „1“ the value of all bits to a defined status. If no parity is set, the parity bit is set to "1", but not evaluated.

0: NONE 1: ODD 2: EVEN

StopBits

The stop bits are set at the end of each transferred character and mark the end of a character.

1: 1Bit 2: 1.5Bit 3: 2Bit

FlowControl

The parameter FlowControl is ignored. When sending RST=0, when receiving RST=1.

RetVal SFC 216 (Error message SER_CFG)

Return values send by the block:

Error code	Description
0000h	no error
809Ah	interface not found
8x24h	Error at SFC-Parameter x, with x: 1: Error at "Protocol" 2: Error at "Parameter" 3: Error at "Baudrate" 4: Error at "CharLength" 5: Error at "Parity" 6: Error at "StopBits" 7: Error at "FlowControl" (Parameter missing)
809xh	Error in SFC parameter value x, where x: 1: Error at "Protocol" 3: Error at "Baudrate" 4: Error at "CharLength" 5: Error at "Parity" 6: Error at "StopBits" 7: Error at "FlowControl"
8092h	Access error in parameter DB (DB too short)
828xh	Error in parameter x of DB parameter, where x: 1: Error 1. parameter 2: Error 2. parameter ...

7.5 Communication

Overview

The communication happens via the send and receive blocks SFC 217 (SER_SND) and SFC 218 (SER_RCV).

The SFCs are included in the consignment of the CPU.

SFC 217 (SER_SND)

This block sends data via the serial interface.

The repeated call of the SFC 217 SER_SND delivers a return value for 3964R, USS and Modbus via RetVal that contains, among other things, recent information about the acknowledgement of the partner station.

USS and Modbus protocols require evaluation of the receipt telegram by calling the SFC 218 SER_RCV after SER_SND.

Parameter

Name	Declaration	Type	Comment
DataPtr	IN	ANY	Pointer to Data Buffer for sending data
DataLen	OUT	WORD	Length of data sent
RetVal	OUT	WORD	Return value (0 = OK)

DataPtr

Here you define a range of the type Pointer for the send buffer where the data that has to be sent is stored. You have to set type, start and length.

Example: Data is stored in DB5 starting at 0.0 with a length of 124Byte.
DataPtr:=P#DB5.DBX0.0 BYTE 124

DataLen

Word where the number of sent bytes is stored.

At **ASCII** if data were sent by means of SFC 217 faster to the serial interface than the interface sends, the length of data to send could differ from the *DataLen* due to a buffer overflow. This should be considered by the user program.

With **STX/ETX**, **3964R**, **Modbus** and **USS** always the length set in DataPtr is stored or 0.

RetVal SFC 217 (Error message SER_SND)

Return values of the block:

Error code	Description
0000h	Send data - ready
1000h	Nothing sent (data length 0)
20xxh	Protocol executed error free with xx bit pattern for diagnosis
7001h	Data is stored in internal buffer - active (busy)
7002h	Transfer - active
80xxh	Protocol executed with errors with xx bit pattern for diagnosis (no acknowledgement by partner)
90xxh	Protocol not executed with xx bit pattern for diagnosis (no acknowledgement by partner)
8x24h	Error in SFC parameter x, where x: 1: Error in "DataPtr" 2: Error in "DataLen"
8122h	Error in parameter "DataPtr" (e.g. DB too short)
807Fh	Internal error
809Ah	Interface not found or interface is used for Profibus
809Bh	Interface not configured

Protocol specific RetVal values

ASCII

Value	Description
9000h	Buffer overflow (no data send)
9002h	Data too short (0Byte)

STX/ETX

Value	Description
9000h	Buffer overflow (no data send)
9001h	Data too long (>1024Byte)
9002h	Data too short (0Byte)
9004h	Character not allowed

3964R

Value	Description
2000h	Send ready without error
80FFh	NAK received - error in communication
80FEh	Data transfer without acknowledgement of partner or error at acknowledgement
9000h	Buffer overflow (no data send)
9001h	Data too long (>1024Byte)
9002h	Data too short (0Byte)

USS

Error code	Description
2000h	Send ready without error
8080h	Receive buffer overflow (no space for receipt)
8090h	Acknowledgement delay time exceeded
80F0h	Wrong checksum in respond
80FEh	Wrong start sign in respond
80FFh	Wrong slave address in respond
9000h	Buffer overflow (no data send)
9001h	Data too long (>1024Byte)
9002h	Data too short (<2Byte)

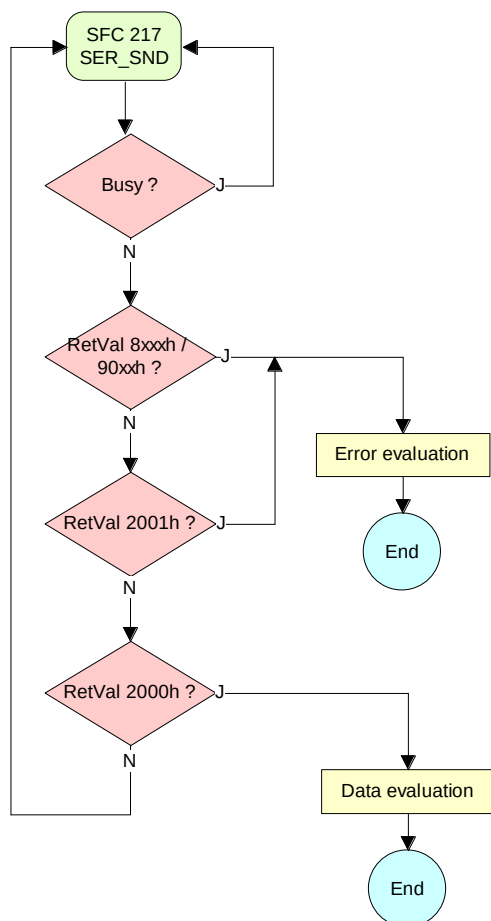
Modbus RTU/ASCII Master

Error code	Description
2000h	Send ready (positive slave respond)
2001h	Send ready (negative slave respond)
8080h	Receive buffer overflow (no space for receipt)
8090h	Acknowledgement delay time exceeded
80F0h	Wrong checksum in respond
80FDh	Length of respond too long
80FEh	Wrong function code in respond
80FFh	Wrong slave address in respond
9000h	Buffer overflow (no data send)
9001h	Data too long (>1024Byte)
9002h	Data too short (<2Byte)

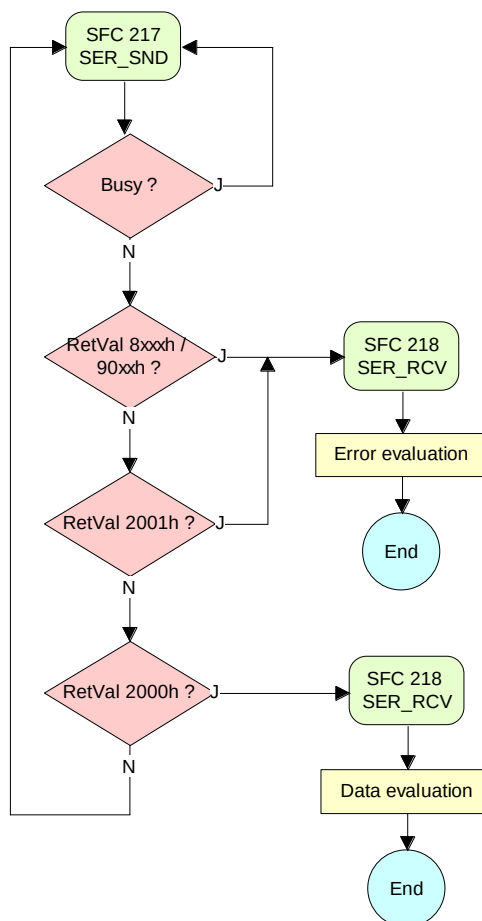
Principles of programming

The following text shortly illustrates the structure of programming a send command for the different protocols.

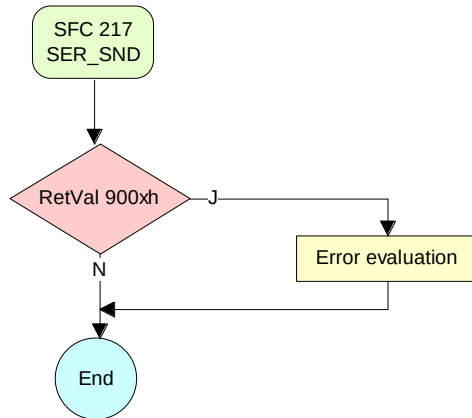
3964R



USS / Modbus



ASCII / STX/ETX



SFC 218 (SER_RCV)

This block receives data via the serial interface.

Using the SFC 218 SER_RCV after SER_SND with the protocols USS and Modbus the acknowledgement telegram can be read.

Parameter

Name	Declaration	Type	Comment
DataPtr	IN	ANY	Pointer to Data Buffer for received data
DataLen	OUT	WORD	Length of received data
Error	OUT	WORD	Error Number
RetVal	OUT	WORD	Return value (0 = OK)

DataPtr

Here you set a range of the type Pointer for the receive buffer where the reception data is stored. You have to set type, start and length.

Example: Data is stored in DB5 starting at 0.0 with a length of 124Byte.
 DataPtr:=P#DB5.DBX0.0 BYTE 124

DataLen

Word where the number of received Bytes is stored.

At **STX/ETX** and **3964R**, the length of the received user data or 0 is entered.

At **ASCII**, the number of read characters is entered. This value may be different from the read telegram length.

Error

This word gets an entry in case of an error. The following error messages may be created depending on the protocol:

ASCII

Bit	Error	Description
0	overrun	Overflow, a sign couldn't be read fast enough from the interface
1	framing error	Error that shows that a defined bit frame is not coincident, exceeds the allowed length or contains an additional Bit sequence (Stopbit error)
2	parity	Parity error
3	overflow	Buffer is full

STX/ETX

Bit	Error	Description
0	overflow	The received telegram exceeds the size of the receive buffer.
1	char	A sign outside the range 20h...7Fh has been received.
3	overflow	Buffer is full

3964R / Modbus RTU/ASCII Master

Bit	Error	Description
0	overflow	The received telegram exceeds the size of the receive buffer.

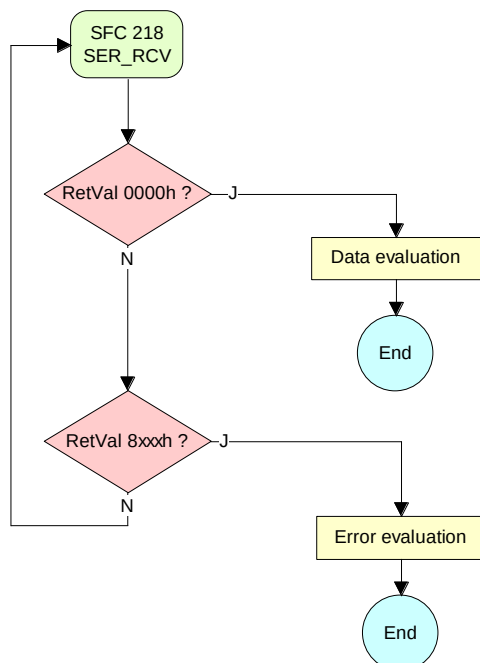
RetVal SFC 218 (Error message SER_RCV)

Return values of the block:

Error code	Description
0000h	no error
1000h	Receive buffer too small (data loss)
8x24h	Error at SFC-Parameter x, with x: 1: Error at "DataPtr" 2: Error at "DataLen" 3: Error at "Error"
8122h	Error in parameter "DataPtr" (e.g. DB too short)
809Ah	Serial interface not found res. interface is used by Profibus
809Bh	Serial interface not configured

Principles of programming

The following picture shows the basic structure for programming a receive command. This structure can be used for all protocols.



8 Profibus Communication

8.1 Overview

Profibus-DP

Profibus is an international standard applicable to an open and serial fieldbus for building, manufacturing and process automation that can be used to create a low (sensor-/actuator level) or medium (process level) performance network of programmable logic controllers.

Profibus comprises an assortment of compatible versions. The following details refer to Profibus-DP.

Profibus-DP is a special protocol intended mainly for automation tasks in a manufacturing environment. DP is very fast, offers Plug'n'Play facilities and provides a cost-effective alternative to parallel cabling between PLC and remote I/O. Profibus-DP was designed for high-speed data communication on the sensor actuator level.

The data transfer referred to as "Data Exchange" is cyclical. During one bus cycle, the master reads input values from the slaves and writes output information to the slave.

Profibus address 1 is reserved

Please regard that the Profibus address 1 is reserved for the system. The address 1 should not be used!

DP slave operation

For the deployment in a super-ordinate master system you first have to project your slave system with configured in-/output areas. Afterwards you configure your master system. Assign your slave system to your master system by dragging the "CPU 313C-2DP" from the hardware catalog at *Configured stations* onto the master system, choose your slave system and connect it.

8.2 Profibus DP Slave

Fast introduction

The deployment of the Profibus section as "intelligent" DP slave happens exclusively at master systems that may be configured in the Siemens SIMATIC manager. The following steps are required:

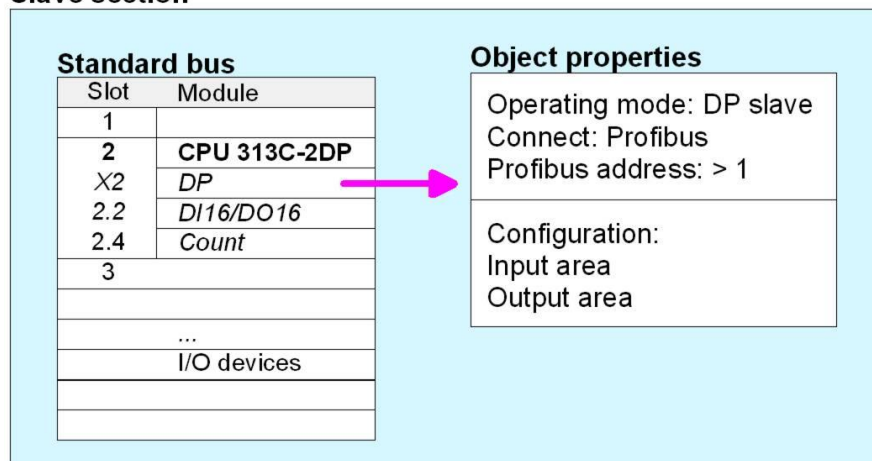
- Start the Siemens SIMATIC manager and configure a CPU 313C-2DP with the operating mode *DP slave*.
- Connect to Profibus and configure the in-/output area for the slave section.
- Save and compile your project.
- Configure another station as CPU 313C-2DP with operating mode *DP master*.
- Connect to *Profibus* and configure the in-/output ranges for the master section
- Save and compile your project.

In the following these steps are more detailed.

Project engineering of the slave section

- Start the Siemens SIMATIC manager with a new project.
- Insert a *SIMATIC 300 station* and name it as "...DP slave".
- Open the hardware configurator and insert a profile rail from the hardware catalog.
- Place the following Siemens CPU at slot 2:
CPU 313C-2DP (6ES7 313-6CF03-0AB0 V2.0)
- Add your modules according to the real hardware assembly.
- Connect the CPU to *Profibus*, set a Profibus address >1 (preferably 3) and switch the Profibus section via *operating mode* to "slave operation".
- Via *Configuration* you define the in-/output address area of the slave CPU that shall be assigned to the DP slave.
- Save and compile your project.

Slave section



Project engineering of the master section

- Insert another *SIMATIC 300 station* and name it as "...DP master".
- Open the hardware configurator and insert a profile rail from the hardware catalog.
- Place the following Siemens CPU at slot 2:
CPU 313C-2DP (6ES7 313-6CF03-0AB0 V2.0)
- Add your modules according to the real hardware assembly.
- Connect the CPU to *Profibus*, set a Profibus address >1 (preferably 2) and switch the Profibus section via *operating mode* to "master operation".
- Connect your slave system to the master system by dragging the "CPU 31x" from the hardware catalog at *configured stations* onto the master system and select your slave system.
- Open the *Configuration* at *Object properties* of your slave system.
- Via double click to the according configuration line you assign the according input address area on the master CPU to the slave output data and the output address area to the slave input data.

- Save, compile and transfer your project. More detailed information about project engineering and project transfer may be found at chapter "Deployment PLC 7001 (CPU 313-6CF03)".

8.3 Profibus installation guidelines

Profibus in general

- A Profibus-DP network may only be built up in linear structure.
- Profibus-DP consists of minimum one segment with at least one master and one slave.
- A master has to be deployed together with a CPU.
- Profibus supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.
- The max. segment length depends on the baud rate:

9.6 ... 187.5kBaud	→	1000m
500kBaud	→	400m
1.5MBaud	→	200m
3 ... 12MBaud	→	100m
- Max. 10 segments may be built up. The segments are connected via repeaters. Every repeater counts for one participant.
- All participants are communicating with the same baudrate. The slaves adjust themselves automatically on the baudrate.
- The bus has to be terminated at both ends.
- Master and slaves can be combined as you like.

Transfer medium

As transfer medium Profibus uses an isolated twisted-pair cable based upon the RS485 interface.

The RS485 interface is working with voltage differences, which is less irritable from influences than a voltage or a current interface. You are able to configure the network as well linear as in a tree structure.

Your SODIMM PLC 7001 module includes an interface which can be connected over additional hardware to a 9pin SubD jack where you connect the Profibus coupler into the Profibus network as a slave.

Max. 32 participants per segment are permitted. The segments are connected via repeaters. The maximum segment length depends on the transfer rate.

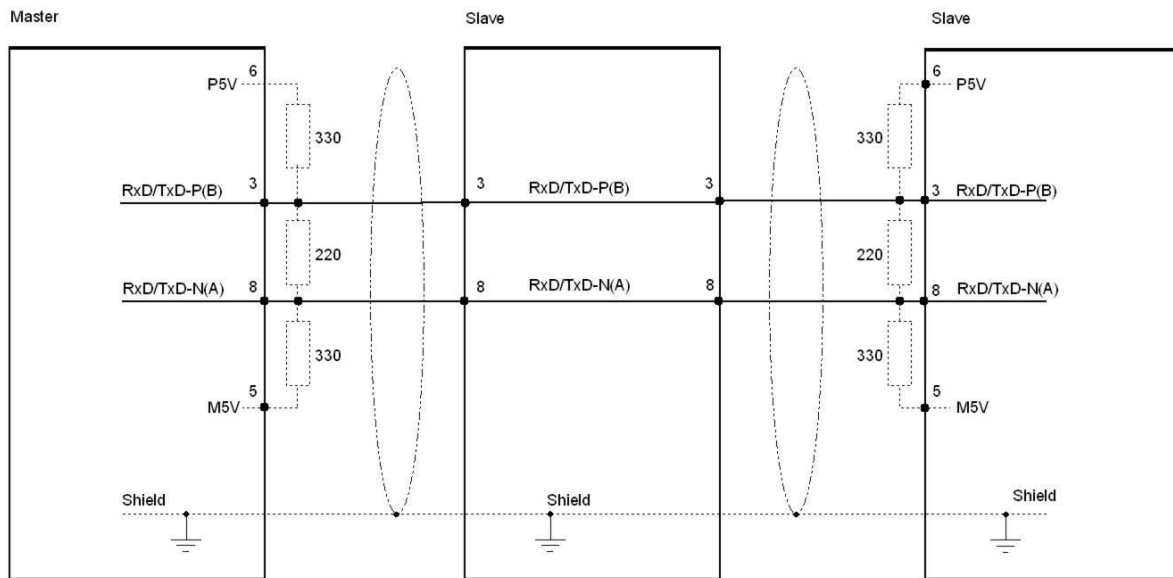
Profibus-DP uses a transfer rate between 9.6kBaud and 12MBaud, the slaves are following automatically. All participants are communicating with the same baudrate.

Profibus Communication

The bus structure under RS485 allows an easy connection res. disconnection of stations as well as starting the system step by step. Later expansions don't have any influence on stations that are already integrated. The system realizes automatically if one partner had a fail down or is new in the network.

Bus connection

The following picture illustrates the terminating resistors of the respective start and end station.



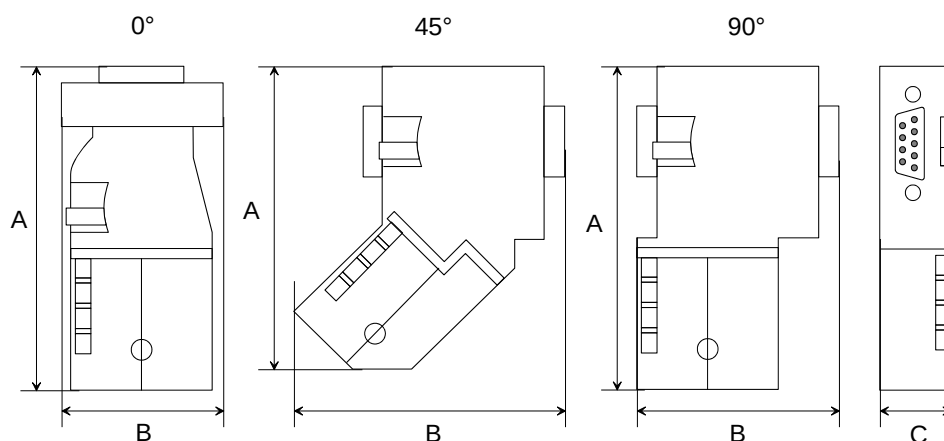
Note!

The Profibus line has to be terminated with its characteristic impedance. Please make sure to terminate the last participants on the bus at both ends by activating the terminating resistor.

"EasyConn" Bus connector

In systems with more than two stations all partners are wired in parallel. For that purpose, the bus cable must be feed-through uninterrupted.

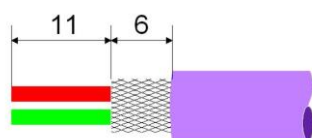
Profichip provides the "EasyConn" connector in different variations (90°, 45° and 0°) with switchable terminating resistor and integrated bus diagnostic. (Order code: PA003105, PA003112, PA003111)



	0°	45°	90°
A	64	61	66
B	34	53	40
C	15.8	15.8	15.8

all in mm

Note!
 To connect this EasyConn plug, please use the standard Profibus cable type A (EN50170) available at Profichip in different length (Order code PA00315x). You may also purchase the "EasyStrip" de-isolating tool from VIPA under the order no. 905-6AA00, which makes the connection of the EasyConn much easier.



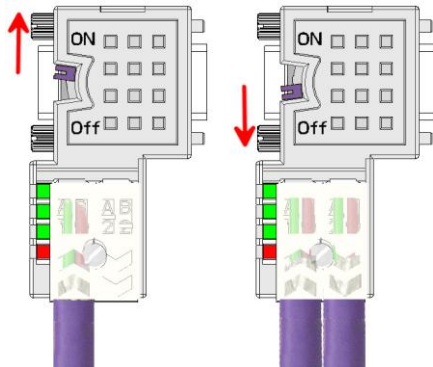
Dimensions in mm



Profibus Communication

Termination with "EasyConn"

The "EasyConn" bus connector is provided with a switch that is used to activate a terminating resistor.



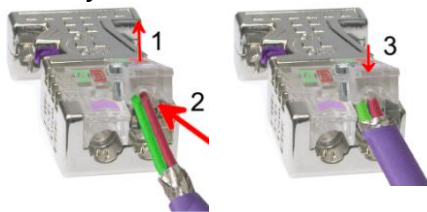
Attention!

The terminating resistor is only effective, if the connector is installed at a slave and the slave is connected to a power supply.

Note!

A complete description of installation and deployment of the terminating resistors is delivered with the connector.

Assembly



- Loosen the screw.
- Lift contact-cover.
- Insert both wires into the ducts provided (watch for the correct line color as below!)
- Please take care not to cause a short circuit between screen and data lines!
- Close the contact cover.
- Tighten screw (max. tightening torque 4Nm).

Please note:

The **green** line must be connected to **A**, the **red** line to **B**!

8.4 Commissioning and Start-up behavior

Start-up on delivery

In delivery the CPU is overall reset. The Profibus part is deactivated.

Online with bus parameter without slave project

The DP master can be served with bus parameters by means of a hardware configuration. As soon as these are transferred the DP master goes online with his bus parameter. This is shown by the RUN LED. Now the DP master can be contacted via Profibus by means of his Profibus address. In this state the CPU can be accessed via Profibus to get configuration and DP slave project.

Slave configuration

If the master has received valid configuration data, he switches to *Data Exchange* with the DP Slaves. This is indicated by the DE-LED.

CPU state controls DP master

After Power ON or a receipt of a new hardware configuration the configuration data and bus parameter were transferred to the DP master.

The DP master does not have any operation switch. His state is controlled by the RUN/STOP state of the CPU.

Dependent on the CPU state the following behavior is shown by the DP master:

Master behavior at CPU RUN

- The global control command "Operate" is sent to the slaves by the master. Here the DE-LED is ON.
- Every connected DP slave is cyclically attended with an output telegram containing recent output data.
- The input data of the DP slaves were cyclically transferred to the input area of the CPU.

Master behavior at CPU STOP

- The global control command "Clear" is sent to the slaves by the master. Here the DE-LED is blinking.
- DP slaves with *fail safe mode* were provided with output telegram length "0".
- DP slaves without *fail safe mode* were provided with the whole output telegram but with output data = 0.
- The input data of the DP slaves were further cyclically transferred to the input area of the CPU.

9 Signal Description

Abbreviations

Abbreviation	Description
PD	Pull Down Resistor
PU	Pull Up Resistor
ST	Schmitt-trigger
LVTTL	Low Voltage transistor-transistor logic level (3.3V)
TTL	transistor-transistor logic level 5V

Complete SODIMM Connector

Pin	Group	Assignment	Dir.	Pin	Group	Assignment	Dir.
1	Supply	VDD1		101	DIO	DO07	O
2		GND1		102		DO04	O
3	MCU	D00	IO	103		DO05	O
4		D01	IO	104		DO02	O
5		D02	IO	105		DO03	O
6		D03	IO	106		DO00	O
7		D04	IO	107		DO01	O
8		D05	IO	108		DI06	I
9		D06	IO	109		DI07	I
10		D07	IO	110		DI04	I
11		D08	IO	111		DI05	I
12		D09	IO	112		DI02	I
13		D10	IO	113		DI03	I
14		D11	IO	114		DI00	I
15		D12	IO	115		DI01	I
16		D13	IO	116	Supply	GND8	
17		D14	IO	117		VDD8	
18		D15	IO	118	MPI	SI_1_TXD	O
19	Supply	VDD2		119		SI_0_TXD	O
20		GND2		120		SI_1_RXD	I
21	MCU	A00	O	121		SI_0_RXD	I
22		A01	O	122		SI_1_RTS	O
23		A02	O	123		SI_0_RTS	O
24		A03	O	124		SI_1_CTS	I
25		A04	O	125		SI_0_CTS	I
26		A05	O	126	Supply	GND9	
27		A06	O	127		VDD9	
28		A07	O	128	USART	SI_3_TXD	O
29		A08	O	129		SI_2_TXD	O
30		A09	O	130		SI_3_RXD	I
31		A10	O	131		SI_2_RXD	I

Pin	Group	Assignment	Dir.	Pin	Group	Assignment	Dir.
32		A11	O	132		SI_3_RTS	O
33		A12	O	133		SI_2_RTS	O
34		A13	O	134		SI_3_CTS	I
35	Supply	VDD3		135	Supply	SI_2_CTS	I
36		GND3		136		GND10	
37	MCU	OE#	O	137	Ethernet	VDD10	
38		WE#	O	138		ETH2_LNK/ACT#	n.c.
39		INT0#	I	139		ETH1_LNK/ACT#	O
40		INT1#	n.c.	140		ETH2_SPEED#	n.c.
41		BS1#	O	141		ETH1_SPEED#	O
42	Supply	WAIT#	I	142		ETH2_TX+	n.c.
43		VDD4		143		ETH1_TX+	O
44	VBUS	GND4		144		ETH2_TX-	n.c.
45		VB_CLK	O	145		ETH1_TX-	O
46		VBUS_DO	n.c.	146		ETH2_AVDD	n.c.
47		VB_DIO (DI)	IO	147		ETH1_AVDD	O
48		VBUS_DO_ENA	n.c.	148		ETH2_RX+	n.c.
49		VB_COM#	O	149		ETH1_RX+	I
50		VB_RDY#	I	150		ETH2_RX-	n.c.
51		VB_LAT#	O	151		ETH1_RX-	I
52	Supply	VB_ARM#	I	152	Supply	GND11	
53		VB_BASP	n.c.	153		VDD11	
54	Supply	GND5		154	FPGA	FPGA_CCLK	O
55		VDD5		155	MMC	MMC_CLK	O
56	DIO	DI30	I	156	FPGA	FPGA_DATA	O
57		DI31	I	157	MMC	MMC_CMD	O
58		DI28	I	158	FPGA	FPGA_PRG#	O
59		DI29	I	159	MMC	MMC_DAT0	IO
60		DI26	I	160	FPGA	FPGA_INIT#	I
61		DI27	I	161	MMC	MMC_DAT1	IO
62		DI24	I	162	FPGA	FPGA_DONE	I
63		DI25	I	163	MMC	MMC_DAT2	IO
64		DO22	I	164	I2C	I2C_CLK	IO
65		DO23	I	165	MMC	MMC_DAT3	IO
66		DO20	O	166	I2C	I2C_DATA	IO
67		DO21	O	167	USB	USB_VCC	I
68		DO18	O	168	Supply	GND12	
69		DO19	O	169	USB	USB_DM	IO
70		DO16	O	170	SYS	RESET_IN#	I
71	Supply	DO17	O	171	USB	USB_DP	IO
72		GND6		172	SYS	SINGLE_BP4	I
73	DIO	VDD6		173	Supply	VDD12	
74		DI22	I	174	SYS	RESET_OUT#	O
75		DI23	I	175		UBAT	
76		DI20	I	176		RUN/STOP#	O
77		DI21	I	177		ANLAUF#	I
78		DI18	I	178	Supply	GND13	

Signal Description

Pin	Group	Assignment	Dir.	Pin	Group	Assignment	Dir.
79		DI19	I	179	SYS	PWRDN#	I
80		DI16	I	180	GPIO	LED_RUN	O
81		DI17	I	181	SYS	BATF#	I
82		DO14	O	182	GPIO	LED_STOP	O
83		DO15	O	183	Supply	VDD13	
84		DO12	O	184	GPIO	LED_SF	O
85		DO13	O	185		GPIO9	n.c.
86		DO10	O	186		LED_FRC	O
87		DO11	O	187		GPIO10	n.c.
88		DO8	O	188	Supply	GND14	
89		DO9	O	189	GPIO	GPIO11	n.c.
90		DI14	I	190		LED_MMC	O
91		DI15	I	191		SPI_CS#	O
92		DI12	I	192		SWITCH_MRES	I
93		DI13	I	193		SPI_CLK	O
94		DI10	I	194		SWITCH_RUN	I
95		DI11	I	195		SPI_SI	I
96		DI08	I	196		GPIO7	n.c.
97		DI09	I	197		SPI_SO	O
98	Supply	GND7		198		GPIO08	n.c.
99		VDD7		199	Supply	VDD14	
100	DIO	DO06	O	200		GND15	

Note: Please see the subsequent chapters for details about the different interfaces.

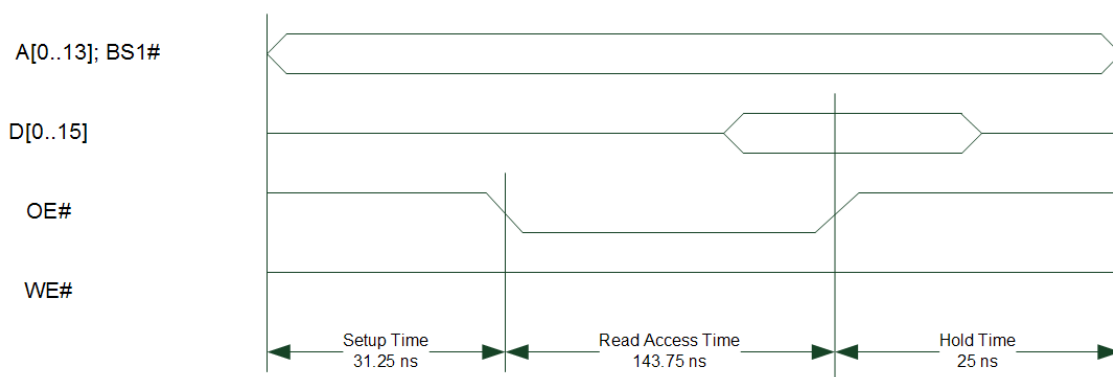
9.1 MCU Interface (Dual-Port-RAM Interface)

Signal	Type	I/O Standard	Characteristics	Function
D[15:0]	IO	LVTTTL 3.3V	24 mA	MCU data bus
A[13:0]	O	LVTTTL 3.3V	24 mA	MCU address bus
OE#	O	LVTTTL 3.3V	24 mA	MCU output enable (READ#)
WE#	O	LVTTTL 3.3V	24 mA	MCU write enable
INT0#	I	LVTTTL 3.3V	Internal PullUp 5V	Reserved for future use. Please leave unconnected.
INT1#	I	LVTTTL 3.3V	External PullUp 5V	External Pullup required! Reserved for future use. Please leave unconnected.
BS1#	O	LVTTTL 3.3V	24 mA	MCU upper byte select
WAIT#	I	LVTTTL 3.3V	Internal PullUP 3.3V	Reserved for future use. Please leave unconnected.

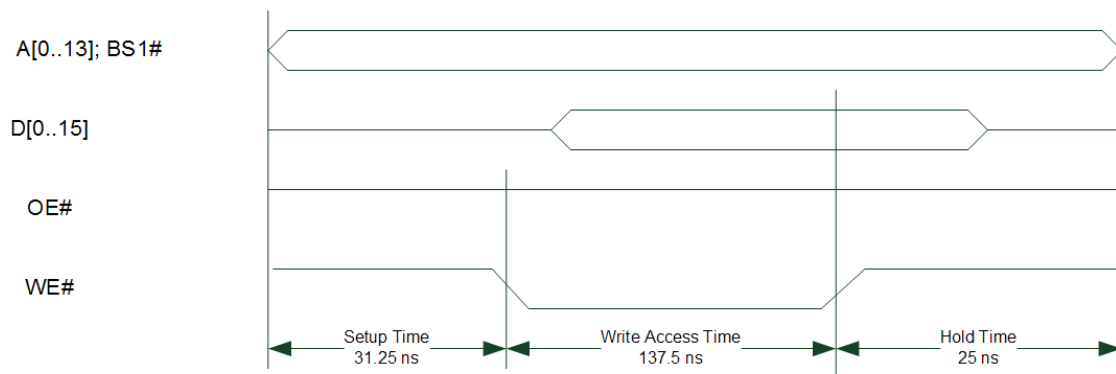
The 16 bit MCU interface serves as a Dual-Port-RAM (DPR) interface to extend the amount of I/O data visible to the PLC program. It can be used to access user specific hardware like analog channels or additional communication interfaces by read and write commands directly from inside the PLC program. The size of the DPR interface is 4096 bytes input data and 4096 bytes output data. In order to decouple the PLC cycle from the update cycle of the user data an external SRAM or FPGA with dual-ported RAM functionality is required. For more information see Chapter “Dual Port Memory Interface”

Timinigs for Dual Port RAM Interface

Read Access



Write Access



9.2 VBUS Interface (Serial Extension Bus)

Signal	Type	I/O Standard	Characteristics	Function
VB_CLK	O	TTL 5V	8 mA	Serial Extension Bus compatible with VIPA System 200V modules.
VBUS_DO	O	LVTTL 3.3V	Don't Use! Pull Up 3.3V	
VB_DIO	IO	TTL 5V	PU(4k7), 8 mA	
VBUS_DO_ENA	O	LVTTL 3.3V	Pull Up 3.3V (reserved)	
VB_COM#	O	TTL 5V	8 mA	
VB_RDY#	I	TTL 5V	PullUp 5V (4k7)	
VB_LAT#	O	TTL 5V	8 mA	
VB_ARM#	I	TTL 5V	PullUp 5V(4k7)	
VB_BASP	n.c.		Not Connected (reserved for future use) Use RUN/STOP# Signal	

The VBUS interface can be used to extend the on-board I/O functions of the SODIMM-PLC7001 by using pre-configured hardware modules of the VIPA System 200V. For more details about which types of modules are supported please contact the profichip support team.

The VBUS interface signals need to be connected to a 9 pin SubD connector (female). An adapter cable is available to connect this SubD connector with the backplane of the VIPA System 200V.

9.3 Digital Inputs (DI)

Signal	Type	I/O Standard	Characteristics	Function
DI[31:16]	I	LVTTL 3.3V	ST, PD(75k)	Digital inputs (not used in this CPU configuration)
DI[15:0]	I	LVTTL 3.3V	ST, PD(75k)	Digital inputs

The DI[15:0] pins serve as digital inputs equipped with configurable input filters and can be read directly from inside the PLC program. In addition they can be assigned with alarm properties and parameterized as counters. Please see the corresponding chapter for a detailed description about the various input functions.

9.4 Digital Outputs (DO)

Signal	Type	I/O Standard	Characteristics	Function
DO[23:16]	O	LVTTL 3.3V	8 mA	Digital outputs (not used in this CPU configuration)
DO[15:0]	O	LVTTL 3.3V	8 mA	Digital outputs

The DO[15:0] pins are digital outputs which can be set and cleared by the PLC program. When technology functions are used some pins have special functions. Please see the corresponding chapters for more details.

9.5 Serial Interfaces (SI_0 - SI_3)

Signal	Type	I/O Standard	Characteristics	Function	
SI_0_TXD	O	LVTTL 3.3V		Transmit data	MPI Interface
SI_0_RXD	I	LVTTL 3.3V		Receive data	
SI_0_RTS	O	LVTTL 3.3V		Ready to send	
SI_0_CTS	I	LVTTL 3.3V		Clear to send	
SI_1_TXD	O	LVTTL 3.3V		Transmit data	Profibus-DP Slave Interface
SI_1_RXD	I	LVTTL 3.3V		Receive data	
SI_1_RTS	O	LVTTL 3.3V		Ready to send	
SI_1_CTS	I	LVTTL 3.3V		Clear to send	
SI_2_TXD	O	LVTTL 3.3V		Transmit data	UART (PtP) Interface
SI_2_RXD	I	LVTTL 3.3V		Receive data	
SI_2_RTS	O	LVTTL 3.3V		Ready to send	
SI_2_CTS	I	LVTTL 3.3V		Clear to send	
SI_3_TXD	O	LVTTL 3.3V		Transmit data	Reserved for future use. Please leave unconnected.
SI_3_RXD	I	LVTTL 3.3V		Receive data	
SI_3_RTS	O	LVTTL 3.3V		Ready to send	
SI_3_CTS	I	LVTTL 3.3V		Clear to send	

The serial interfaces of the SODIMM-PLC7001 are providing dedicated communication functions like mentioned in the table above. Please see the chapter “External Circuit Description” for details about how to connect the pins of the SODIMM module to the associated physical layers.

9.6 Ethernet Interfaces (ETH1 & ETH2)

Signal	Type	I/O Standard	Characteristics	Function
ETH1_TX+	IO	ETH PHY		Physical transmit pair (+ differential)
ETH1_TX-	IO	ETH PHY		Physical transmit pair (- differential)
ETH1_RX+	IO	ETH PHY		Physical receive pair (+ differential)
ETH1_RX-	IO	ETH PHY		Physical receive pair (- differential)
ETH1_LNK/ACT#	O	LVTTTL 3.3V	8 mA	Link/Activity LED OFF (H): No Link ON (L): Link TOGGLE: Activity
ETH1_SPEED#	O	LVTTTL 3.3V	8 mA	Speed LED OFF (H): 10BT ON (L): 100BT
ETH1_AVDD				3.3 V supply for RJ45 jack
ETH2_TX+	n.c.			ETH2 not used in current configuration.
ETH2_TX-	n.c.			
ETH2_RX+	n.c.			
ETH2_RX-	n.c.			
ETH2_LNK/ACT#	n.c.			
ETH2_SPEED#	n.c.			
ETH2_AVDD	n.c.			

The Ethernet PHY is integrated on the SODIMM-PLC7001 module. The Ethernet signals can be connected to the corresponding pins of a RJ45 jack directly. Please see the chapter “External Circuit Description” for details.

9.7 FPGA Configuration Interface

Signal	Type	I/O Standard	Characteristics	Function
FPGA_CCLK	O	LVTTTL 3.3V	8 mA	Do not connect (reserved for future use!)
FPGA_DATA	O	LVTTTL 3.3V	8 mA	Do not connect (reserved for future use!)
FPGA_PRG#	O	LVTTTL 3.3V	8 mA	Do not connect (reserved for future use!)
FPGA_INIT#	I	LVTTTL 3.3V	ST	Do not connect (reserved for future use!)
FPGA_DONE	I	LVTTTL 3.3V	ST	Do not connect (reserved for future use!)

9.8 Multi-Media-Card Interface (MMC)

Signal	Type	I/O Standard	Characteristics	Function
MMC_CLK	O	LVTTL 3.3V	8 mA	Clock of an MMC or SD Card
MMC_CMD	IO	LVTTL 3.3V	8 mA	Command/Response of an MMC or SD Card
MMC_DATA[3:0]	IO	LVTTL 3.3V	8 mA	Data[0] of an MMC Data[3:0] of an SD Card

This interface is intended to be used with standard Multi-Media-Cards to store PLC program data permanently. In addition specially prepared Memory Configuration Cards (MCC) are available to extend the internal PLC memory.

The MultiMedia Card Interface supports the MultiMediaCard (MMC) Specification V2.2 and the port signals can be directly connected to an appropriate MMC socket. Please see the chapter "External Circuit Description" for details.

9.9 USB Interface

Signal	Type	I/O Standard	Characteristics	Function
USB_DM	IO	USB Transceiver	8 mA	Reserved for future use. Please leave unconnected.
USB_DP	IO			Reserved for future use. Please leave unconnected.
USB_VCC	I	LVTTL 3.3V	PD(10k)	Reserved for future use. Please leave unconnected.

USB 2.0 Full Speed (12 Mbits per second) Device Port with integrated transceiver. The port signals can be connected directly to an appropriate USB socket. Please see the chapter "External Circuit Description" for details.

The current configuration does not support USB communication.

9.10 I²C Interface

Signal	Type	I/O Standard	Characteristics	Function
I2C_CLK	IO	LVTTL 3.3V	8 mA, PU(1k)	Reserved for future use. Please leave unconnected.
I2C_SDA	IO	LVTTL 3.3V	8 mA, PU(1k)	Reserved for future use. Please leave unconnected.

High-speed mode compatible I²C interface which can be used to connect to external devices like EEPROMs or ADCs and DACs supporting I²C communication.

The current configuration does not support I²C communication.

9.11 System Pins

Signal	Type	I/O Standard	Characteristics	Function
RESET_IN#	I	LVTTL 3.3V	PU(4k7)	Master Reset Input (must be held low for at least 1 second for correct startup)
RESET_OUT#	O	LVTTL 3.3V	8 mA	Master Reset Output
RUN/STOP#	O	LVTTL 3.3V	8 mA, PD(1k)	Operational Status of the CPU: this signal is high (Run) after OB100 is finished and during normal operation Cycle (OB1/OB35...), low in every fault condition and during Stop
ANLAUF#	I	LVTTL 3.3V	PD(1k)	Reserved for future use. Please leave unconnected.
PWRDN#	I	LVTTL 3.3V	PU(1k)	Power Down (Power Fail): This signal should change its state from high to low when the PLC-System power supply detects powerloss. IMPORTANT: The power supply MUST provide stable power for at least 3 miliseconds (ms) AFTER PWRDN# goes low! Keep this signal high if not used!
BATF#	I	LVTTL 3.3V	PU(1k)	Battery Fail: this signal is sampled after startup and provides information about the battery for the retention function of the Low Power SRAM. If low: There is not enough power, If high: All ok!
SINGLE_BP_4	I	LVTTL 3.3V	PU(1k)	Reserved for future use. Please leave unconnected.
UBAT	I	3.0V	Max. 100µA Typ. 30µA	Supply voltage for the Low Power SRAM to provide the memory retention function in case of power loss

Comments on the memory retention function:

The memory retention function prevents the loss of data and instructions stored in low power SRAMs (all OB/FC/FB/DBs and all markers and stacks). There are two functions to ensure the correct behavior: a CRC checksum which is generated in the 3 ms after PWRDN# signal goes low and the BATF Signal.

The PLC does a master reset when the CRC Check results in an error during the next startup or the BATF signal is low. If a valid program file is detected on the MMC/MCC, this file will be transferred into the work memory. To ensure that the PLC executes the user program a BATF-handling block (OB83) is necessary on the MMC/MCC. In addition a BATF-message will be created in the diagnostic buffer.

Leave UBAT, PWRDN# and BATF# unconnected and attach a valid MMC/MCC when no retention function is necessary.

9.12 General Purpose IOs (GPIO)

Signal	Type	I/O Standard	Characteristics	Function
GPIO0	O	LVTTL 3.3V	8 mA	LED Run
GPIO1	O	LVTTL 3.3V	8 mA	LED Stop
GPIO2	O	LVTTL 3.3V	8 mA	LED SF (Global Error)
GPIO3	O	LVTTL 3.3V	8 mA	LED FRCE (Force)
GPIO4	O	LVTTL 3.3V	8 mA	LED MMC (MMC Activity)
GPIO5	I	LVTTL 3.3V	PU(4k7)	Switch MRES (Master Reset, Overall Reset)
GPIO6	I	LVTTL 3.3V	PU(4k7)	Switch RUN/STOP
GPIO7	n.c.			Reserved for future use. Please leave unconnected.
GPIO8	n.c.			
GPIO9	n.c.			
GPIO10	n.c.			
GPIO11	n.c.			
GPIO12	n.c.			
GPIO13	n.c.			
GPIO14	n.c.			
GPIO15	n.c.			

10 Electrical Characteristics

10.1.1 Absolute Maximum Ratings

Symbol	Parameter	Conditions	Min	Max	Unit
V _{DD}	DC Supply Voltage		-0.3	6.0	V
V _I	Input Voltage	LVTTL 3.3V	-0.3	3.9	V
		TTL 5V	-0.3	5.8	V
V _O	Output Voltage	LVTTL 3.3V	-0.3	3.9	V
		TTL 5V	-0.3	V _{DD} + 0.3	V
I _O	DC Output Current	see IO tables for details	-24	24	mA
T _{Store}	Storage Temperature		-40	+125	°C

10.1.2 Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V _{DD}	DC Supply Voltage		4.5	5.5	V
GND	Circuit Ground		0	0	V
V _I	Input Voltage	LVTTL 3.3V	0	3.3	V
		TTL 5V	0	5.0	V
T _A	Ambient Temperature		0	+70	°C

10.1.3 DC Characteristics of Standard IOs

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V _{DD}	DC Supply Voltage		4.5	5.0	5.5	V
V _{IL}	Input Low-level Voltage	LVTTL 3.3V, TTL 5V			0.8	V
V _{IH}	Input High-level Voltage	LVTTL 3.3V, TTL 5V	2.0			V
V _{T-}	Schmitt-Trigger negative going Threshold Level	LVTTL 3.3V, ST	0.8	1.1		V
V _{T+}	Schmitt-Trigger positive going Threshold Level	LVTTL 3.3V, ST		1.6	2.0	V
V _{OL}	Output Low-Level Voltage	LVTTL 3.3V, TTL 5V			0.4	V
V _{OH}	Output High-Level Voltage	LVTTL 3.3V	2.4		3.3	V
		TTL 5V	2.4		V _{DD}	V
C _{IN}	Input Capacitance				20	pF
I _{IN}	Input Leakage Current	V _{IN} = 3.3V or 0V	-10	±1	+10	µA
I _O	Output Current	Please see corresponding tables for details.				

10.1.4 DC Characteristics of USB Transceiver

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Input Low-level Voltage	USB Transceiver			0.8	V
V_{IH}	Input High-level Voltage	USB Transceiver	2.0			V
V_{DI}	Differential Input Sensitivity	$ (D+) - (D-) $	0.2			V
V_{CM}	Differential Common Mode Range		0.8		2.5	V
V_{OL}	Output Low-level Voltage	USB Transceiver			0.3	V
V_{OH}	Output High-level Voltage	USB Transceiver	2.8			V
C_{IN}	Transceiver Capacitance				20	pF
I_{IN}	Input Leakage Current	$0V < V_{IN} < 3.3V$	-5		+5	μA
I_O	Output Current			8 mA		μA

10.1.5 DC Characteristics of Ethernet PHY

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_O	Peak Differential Output Voltage	100Base-TX	0.95		1.05	V
V_{IMB}	Output Voltage Imbalance	100Base-TX			2	%
V_P	Peak Differential Output Voltage	10Base-T	2.2		2.8	V

11 External Circuit Description

This chapter describes circuit schematics in order to use the signals accessible at the module in standard conform ways such as connecting to a jack. The modules and elements used in the schematics may be exchanged by analog modules from other producers.

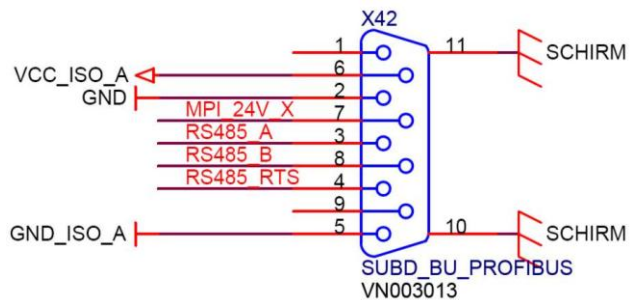
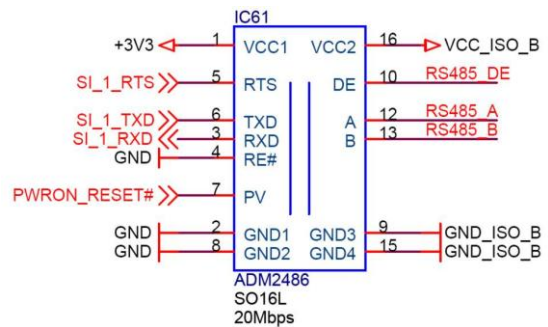
11.1 SubD jack 9pin

The SubD 9 pin jack is the standard connector for Profibus, MPI and PtP connections. In order to connect the SODIMM PLC 7001 module to this standard you need additional hardware. The 5V supply voltage for the interfaces is recommended to be electrically isolated. The signal lines should be DC-isolated by the used periphery.

There fore a high speed, half-duplex isolated RS485 Transceiver is recommended. The picture shows the one in use at the EVABoard for the SODIMM module. The power supply and pin assignment is chip specific so in case you use a different device please refer to the manual from the producer. The bus side voltage however must be an isolated 5V supply in order to match bus standard.

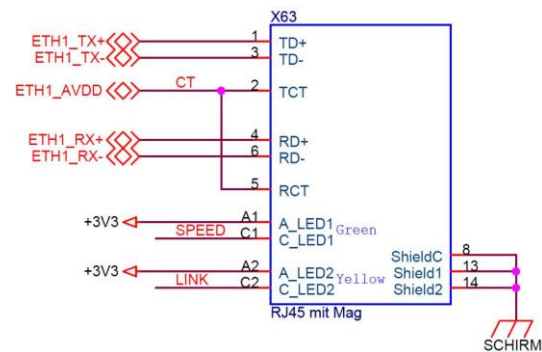
An isolated DC-DC-converter may be extended by a voltage stabilizer. This is depending on quality of your converter.

The pin assignment of a standard 9 pin SubD jack is shown in this picture.



11.2 RJ45 Ethernet jack

To connect a RJ45 Ethernet jack no additional hardware is required except the LEDs for speed and link need a series resistor.



11.3 USB jack

Like the RJ45 jack the USB jack also requires no additional hardware to be connected to the SODIMM-module. Please refer to the manual of the jack to get the Pin information you need.

11.4 MMC Card-slot

The MMC Card-slot may be directly connected to the pins of the PLC 7001 module. There is no additional hardware required. Please refer to the manual of your producer.

11.5 LEDs of the GPIO interface

The LEDs of the GPIO interface are all considered as low active. In order to get well known signals from your SODIMM module (e. g. for overall reset) it is recommended that you don't change that. In addition you will need series resistors for voltage correction.

Colors of Status LEDs

Signal	Color
Power	green
RUN	green
STOP	yellow
SF	red
FRCE	yellow
MMC	yellow

Appendix

A – Index

3964R	95	Memory management.....	11
Applications	18	minimal project.....	25
ASCII	94	MMC.....	38, 123
Battery buffer	23	MMC card-slot.....	130
Breakpoints.....	41	MMC-Cmd	
Communication		Auto commands.....	51
RS485	12	Modbus	97
Profibus	13	Operands	19
PtP	13	Operating mode	40
Compatibility	21	operating structure	18
Counter		Overall reset.....	42
Address assignment	14, 60	Factory setting	44
CPU	17	Overview	
CPU parameterization	30	CPU 313SC/DPM.....	20
Cycle time surveillance.....	42	Parameterization	
Deployment		Digital part	15, 61
I/O peripheral	9, 59	Modules	35
PtP communication	93	PLC functions.....	57
DI	120	Power supply	21
Diagnostic		Procedures.....	95
Buffer.....	52	processing.....	18
MPI	38	Profibus DP	
Digital part.....	14, 60	Baud rate	111
Address assignment	14, 60	Commissioning	115
DO	121	Connectors	112
Environmental conditions	20	dismantling lengths.....	113
Ethernet	122, 130	Installation guidelines.....	111
Ethernet PG/OP	24	Line termination	114
Event-ID.....	52	Start-up behavior	115
Factory setting	44, 47	Transfer medium	111
Firmware.....	44	Profibus DP master	
FPGA	122	Deployment as DP slave	109
GPIO	125	Overview.....	109
I2C	123	project engineering	13, 59
In-/Output range	13, 60	Project engineering	27
Interfaces.....	12	Transfer	37
Ethernet PG/OP	11	Ethernet PG/OP channel	39
MPI	12	MMC	38
Profibus	12	MPI	37
PtP	12	Diagnostics.....	38
RS485		Project engineering CPU	28
Profibus	13	Project engineering Ethernet	28
PtP	13, 93	PtP communication	93
Know-how protection	48	3964R	95
LEDs	11, 130	ASCII	94
MCC.....	11, 47	Broadcast	96
MCU.....	118	Communication.....	102
Memory extension	47		

Error messages.....	102, 103, 107	Status LEDs.....	130
Fast introduction	93	STX/ETX	94
Modbus.....	97	SubD jack.....	129
Parameterization.....	100	Supported parameters	30
Principals data transfer	98	System	124
Protocols	94	Test functions.....	56
RS485 interface	99	timer and counter	19
SFCs	93, 102	USB.....	123
STX/ETX	94	USB jack	130
USS	96	USS.....	96
RJ45 jack.....	130	VBUS	120
Serial.....	121	Watchdog.....	42
Speed7 200V.....	29, 34	Web page.....	26
Start-up behavior	23	wld files	38, 49

Revision History

Version	Date	Page	Remarks
V1.00	14.05.2009		First release
V1.01	12.07.2010	Chap. 9 Chap. 6	comments to signal descriptions added chapter "Dual Port Memory Interface" added

profichip GmbH
Einsteinstrasse 6
91074 Herzogenaurach
Germany

Phone : +49.9132.744-200
Fax: +49.9132.744-2064

www.profichip.com

